

UNIVERSIDAD NACIONAL DE INGENIERÍA



Monografía para optar a título de Ingeniero en Computación

Sistema web para la Planeación Operativa en las Direcciones del Instituto
Nicaragüense de Estudios Territoriales INETER

Elaborado por:

Br. José Denis Guevara García

Carné:

2013-61364

Tutor: Ing. Adilson González López

Asesor: Ing. Luis Herrera

Fecha: 10 de septiembre del 2018

Contenido

1.	Introducción.....	3
2.	Antecedentes	4
3.	Justificación.....	6
4.	Objetivos	7
4.1.	Objetivos generales	7
4.2.	Objetivos específicos	7
5.	Marco teórico	8
5.1.	Ingeniería de software orientado a objetos	8
5.1.1.	Conceptos de la orientación a objetos.....	8
5.1.2.	Lenguaje Unificado de Modelado (UML)	9
5.1.3.	Herramienta CASE	10
5.1.4.	Diagramas UML.....	11
5.1.5.	Diagrama de casos de uso	11
5.1.6.	Diagrama de clases	12
5.1.7.	Diagrama de comunicación	12
5.2.	Plan Operativo Anual	13
5.3.	Sistemas de información.....	14
5.4.	Base de datos	15
5.5.	Gestor de base de datos.....	15
5.6.	Software.....	16
5.6.1.	Lenguaje de programación	17
5.6.2.	Web socket.....	17
5.6.3.	Redis	18
5.7.	Servidor web	18

5.8.	Aplicación web	19
5.9.	Modelo en espiral.....	20
5.10.	Técnicas de recopilación de información	21
6.	Desarrollo del sistema.....	22
6.1.	Recopilación y análisis de la información.....	23
6.1.1.	Requerimientos generales.....	23
6.1.2.	Reglas del negocio	24
6.1.3.	Requerimientos específicos.....	25
6.1.4.	Estudio de factibilidad.....	27
6.2.	Modelado de requerimientos.....	32
6.2.1.	Modelado de casos de uso	32
6.2.2.	Diagrama de clase.....	49
6.2.3.	Diseño de la base de datos	51
6.2.4.	Diagramas de secuencia	51
6.2.5.	Diagrama de despliegue.....	59
6.3.	Desarrollo del software	60
6.3.1.	Análisis de riesgos y plan de contención	62
6.4.	Pruebas.....	65
6.5.	Implementación.....	68
7.	Evaluación de resultados	73
7.1.	Métricas	73
7.2.	Método de evaluación.....	75
7.2.1.	Procedimiento para evaluar la creación de una estructura de trabajo.....	75
7.3.	Resultados obtenidos para la creación de estructura de trabajo	75
7.3.1.	Evaluación de desempeño	75

7.3.2.	Evaluación de facilidad de uso	76
7.4.	Resultados obtenidos para el reporte de planificación.....	78
7.4.1.	Evaluación de desempeño	78
7.4.2.	Evaluación de la facilidad de uso.....	78
7.5.	Resultados de evaluación de estabilidad	79
8.	Conclusiones y recomendaciones.....	84
8.1.	Conclusiones	84
8.2.	Recomendaciones	84
9.	Anexos	86
9.1.	Anexo 1: Entrevista Dir. Desarrollo	86
9.2.	Anexo 2: Entrevista Dir. Geo repositorio	87
9.3.	Anexo 3: Diseño de la base de datos	88
9.3.1.	Tabla actividades.....	90
9.3.2.	Tabla ar_internal_metadata	91
9.3.3.	Tabla comment_reunions	91
9.3.4.	Tabla comments	92
9.3.5.	Tabla likes	93
9.3.6.	Tabla local_managements.....	94
9.3.7.	Tabla locals	95
9.3.8.	Tabla logged_actions.....	95
9.3.9.	Tabla messages	97
9.3.10.	Tabla notifications	98
9.3.11.	Tabla projects.....	98
9.3.12.	Tabla projects_directions.....	100
9.3.13.	Tabla registration_tokens	101

9.3.14.	Tabla replies.....	102
9.3.15.	Tabla reunions.....	102
9.3.16.	Tabla reunions_users.....	104
9.3.17.	Tabla roles.....	104
9.3.18.	Tabla schema_migrations	106
9.3.19.	Tabla technologies	106
9.3.20.	Tabla users.....	107
9.3.21.	Tabla work_histories.....	109
9.3.22.	Tabla works	110
9.4.	Anexo 4: Cronograma de ejecución.....	111
9.5.	Anexo 5: Ficha de registro de prueba	113
9.6.	Anexo 6: Registro de reuniones.....	114
9.7.	Anexo 7: Versiones de aplicación	117
9.7.1.	Anexo 7.1: vista de tareas versión 1.1	117
9.7.2.	Anexo 7.2: vista de lista de proyectos versión 2.1	118
9.7.3.	Anexo 7.3: vista calendario versión 2.1	119
9.7.4.	Anexo 7.4: formulario editar proyecto versión 2.1.....	120
9.7.5.	Anexo 7.5: vista mostrar proyecto	120
9.7.6.	Anexo 7.6: vista de lista de proyectos versión 2.2	121
9.7.7.	Anexo 7.7: vista de mostrar proyecto versión 2.2	121
9.7.8.	Anexo 7.8: vista de reuniones versión 2.2	122
9.7.9.	Anexo 7.9: vista de calendario versión 3.1	123
9.7.10.	Anexo 7.10: vista de organigrama versión 3.1	123
9.7.11.	Anexo 7.11: vista de mostrar proyecto versión 3.1.....	124
9.7.12.	Anexo 7.12: vista últimos avances registrados versión 3.2	124

9.7.13.	Anexo 7.13: vista de informe	125
9.7.14.	Anexo 7.14: vista validar tareas versión 3.2	127
9.7.15.	Anexo 7.15: vista invalidar tarea versión 3.2	127
9.7.16.	Anexo 7.16: vista validar tarea versión 3.2	128
9.7.17.	Anexo 7.17: vista de mensajes versión 3.3	128
9.7.18.	Anexo 7.18: vista de calendario versión 3.3	129
9.7.19.	Anexo 7.19: vista mostrar proyecto versión 3.3	130
9.7.20.	Anexo 7.20: formulario editar reunión versión 3.3	130
9.7.21.	Anexo 7.21: vista mostrar reunión	131
9.8.	Anexo 8: Plan informe de trabajo DSGI	132
9.9.	Anexo 9: Diagrama físico de red	133
9.10.	Anexo 10: Carta de aceptación	134
10.	Bibliografía	135
11.	Índice de ilustraciones	137
12.	Índices de tablas	139

1. Introducción

En el presente documento se describe el desarrollo de un sistema web para automatizar los procesos administrativos de planificación de proyectos en el Instituto Nicaragüense de Estudios Territoriales INETER, con el fin de mejorar el control de las tareas y actividades, facilitando el procesamiento de la información y la generación de reportes, así como la comunicación entre los equipos de trabajos, contribuyendo de esta manera a los directivos y jefes de proyectos en tomas de decisiones.

En cualquier institución ya sea pública o privada es necesario que exista un plan operativo anual que es el que asegura el cumplimiento o avance de los objetivos de la organización, enfocado a alcanzar los alineamientos de la institución, este también ayuda a mantener un orden en la secuencia de las actividades y definir situaciones problemáticas. En el caso particular (INETER), documento que se entrega a la Procuraduría General de la República anualmente.

El documento presenta el desarrollo de software utilizando la metodología de desarrollo de espiral, comenzando con una fase de modelado de las especificaciones de los interesados por parte de la institución: directores, subdirectores de área, empleados, y como principal interesado el Dr. Vladimir Gutiérrez Corea, co-director de INETER. Se plantea el uso de diagramas de casos de uso, secuencia y clases para definir funcionalidades y alcances necesarios para obtener los mejores resultados de respuesta y usabilidad del sistema.

Revisa el tiempo de redacción.

La presentación de la propuesta está estructurada en este documento de la siguiente manera: descripción de Antecedentes, la Justificación misma del desarrollo, luego los objetivos a seguir, el marco teórico utilizados, metodología de desarrollo, y finalmente las referencias bibliográficas tomadas para la elaboración de este documento.

El sistema almacena la información en el centro de datos de INETER, usando como gestor de base de datos *PostgreSQL*, usando *Redis* como gestor base de datos de caché, permitiendo que la información este en un lugar seguro, sea confiable y esté disponible en cualquier momento que sea necesario.

2. Antecedentes

Para hablar un poco de la estructura organizacional actual, INETER se divide en 7 direcciones generales: geodesia, catastro, meteorología, sistemas geoinformáticos, recursos hídricos, ordenamiento territorial y geodesia, *ver figura 1*. En ellas encontraremos diferentes sub direcciones y los proyectos que se dividen en categorías según su tipo y su finalidad. Su duración es estimada según la complejidad de este, cada proyecto puede tener actividades y tareas a las que se pueden agregar los participantes, quienes van a trabajar en las tareas específicas según el rol que desempeñan dentro del proyecto.

Cada dirección cuenta con diferentes subdirecciones que se dividen por actividad o servicio, el director y el sub director usan la metodología de organigrama vertical para delegar responsabilidades y las personas que están dentro de cada tarea.

Además de proyectos temporales, también se observó que en algunas áreas de trabajo se dedica mayor tiempo a tareas que son repetitivas, por ejemplo: tareas administrativas, contables, de recepción, etc.

El plan operativo anual actualmente se elabora en una hoja de cálculo de Excel, es decir, que cada dirección tiene que desarrollar manualmente el plan y aún no se cuenta con un sistema que automatice y centralice la información referida al plan operativo, resultando en un control poco eficiente. No existe una monitorización que permita brindar en tiempo real las tareas que se están llevando a cabo en las diferentes direcciones de INETER. Esto es una desventaja porque no se lleva un

control de las tareas que se están realizando, cuánto tiempo se lleva una determinada tarea, y el avance de esta.

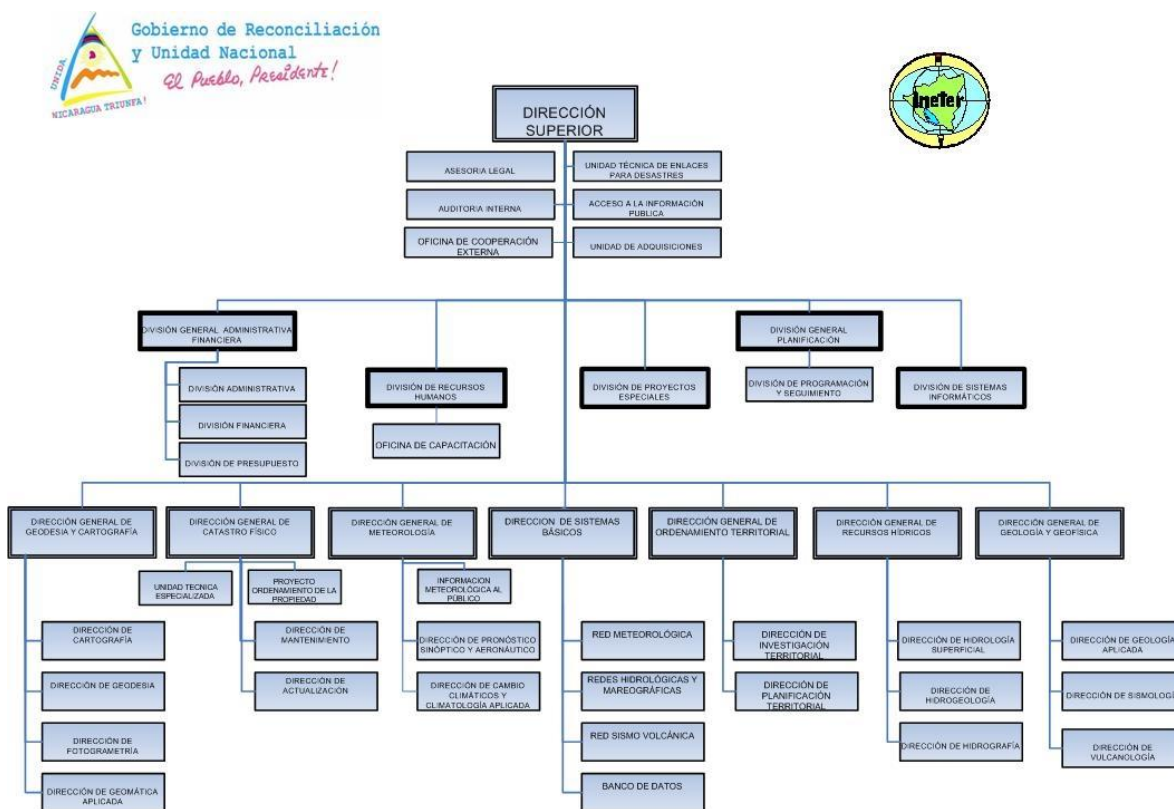


Figura 1 Organigrama de INETER

El plan se realiza al principio del año con base en el presupuesto de la institución y conforme a este se definen los costos de los recursos y servicios que se van a necesitar para completar los proyectos definidos y representan una guía del flujo de trabajo para lograr una mayor eficiencia, eficacia y calidad en los bienes y servicios que se proveen, además de ser una de las principales herramientas en la toma de decisiones a partir de diagnóstico de la situación actual para lograr alcanzar los objetivos deseados. Cabe mencionar que la aplicación no lleva el control de los recursos financieros de la institución.

Es muy importante mencionar que el Sistema Administrativo Financiero (SAF) de la Dirección Administrativa Financiera de INETER es el que lleva el control del flujo de efectivos de ingresos y egresos, la propuesta se basa directamente en el control y planificación de proyectos operativos y tareas generales del personal del instituto.

Sin embargo, se pretende automatizar este proceso para facilitar el control y generación de reportes gráficos disminuyendo el tiempo de este proceso y teniendo un mejor detalle de los proyectos y actividades que se realizan en la institución en el momento que se está llevando a cabo.¹

3. Justificación

Para aportar a este mecanismo de trabajo se planteó crear una aplicación web en la que se puedan registrar los proyectos, actividades y tareas, dar seguimiento a estas a través de los directores y subdirectores encargados de cada área. El personal es el que se va a encargar de registrar los avances de cada día según las tareas que realicen.

El sistema pretende tener centralizada la información, un mejor tiempo de respuesta y optimiza la toma de decisiones a los directores generales y responsables de las subdirecciones en cuanto a los proyectos que se están llevando a cabo, mediante un canal de comunicación directa para los usuarios del sistema, para esto se implementa una aplicación de chat mediante web socket y reportes gráficos que indiquen tiempos y cantidad de tareas que están llevándose a cabo, información que antes no estaba disponible.

Se pretende llevar un mejor control de las tareas que se realizan y una mejora notable en los tiempos de respuesta y de creación de informes importantes para la toma de decisiones de cualquier dirección.

Los directores, subdirectores, y encargados de personal requieren generar reportes gráficos muy necesarios para el análisis y la toma de decisiones, así como la generación de informes que permitan medir en tiempo y esfuerzo invertido para las

¹ Se redactó de información recopilada de entrevistas con los directores y sub-directores del área de sistemas Geo informáticos. Ver *anexo 1 y 2*

tareas dentro de la institución y de paso ayudar también con la conservación ambiental haciendo una reducción en gasto de papelería e impresiones.

4. Objetivos

4.1. Objetivos generales

Desarrollar un sistema web para el seguimiento operativo de los proyectos y tareas que se realizan en las diferentes direcciones de Instituto Nicaragüense de Estudios Territoriales INETER.

4.2. Objetivos específicos

- Analizar la estructura de trabajo y planificación que actualmente utiliza INETER para la planificación operativa mediante las técnicas de recopilación de información.
- Diseñar un sistema de Información web, a partir de los requerimientos establecidos, basado en el lenguaje unificado de modelado (UML).
- Construir el software en base al diseño realizado y cumpliendo con las etapas de desarrollo de software usando como lenguaje de programación Ruby, Rails como framework de desarrollo y postgresSQL como gestor de base de datos.
- Implantar el sistema en los servidores de INETER y recolectar información de su aplicación.

5. Marco teórico

5.1. Ingeniería de software orientado a objetos

Según Lewis, 1994 “software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo” Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software (SE del inglés Software Engineering) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos.²

La Ingeniería de Software Orientada a Objetos, es un enfoque que se centra en el problema basado en una visión orientada a objetos, donde el dominio del problema se caracteriza mediante un conjunto de estos. Tomando cada entidad como una clase y derivando responsabilidades y tareas para cada una de estas.³

5.1.1. Conceptos de la orientación a objetos

Según Pressman, 2010, “para entender el punto de vista orientado a objetos, considere un ejemplo de un objeto del mundo real: el objeto sobre el cual se sienta en este momento, una silla. Silla es una subclase de una clase mucho más grande que puede llamar **PiezaDeMobiliario**. Las sillas individuales son miembros (por lo general llamadas instancias) de la clase silla. Un conjunto de atributos genéricos puede asociarse con cada objeto de la clase **PiezaDeMobiliario**. Por ejemplo, todos

² Lewis. (1994). *What is Software Engineering?* Retrieved from Angelfire.

³ Vladimir Gutierrez, J. U. (2007). Trabajo monográfico. *Universidad Nacional de Ingeniería*.

los muebles tienen un costo, dimensiones, peso, ubicación y color, entre muchos posibles atributos. Lo mismo se aplica si se habla de una mesa, una silla, un sofá o un armario. Puesto que silla es miembro de **PiezaDeMobiliario**, silla hereda todos los atributos definidos para la clase.”⁴

Algunos conceptos que tenemos que tomar en cuenta son:

Clases: Estos objetos también llamados **clases** encapsulan datos, estructuras y abstracciones procedurales requeridos para describir el contenido o comportamiento de alguna entidad del mundo real. Dicho de otra forma, una clase es una descripción generalizada (por ejemplo, una plantilla o plano) de una colección de objetos similares. Ver *anexo 1.b* Por definición, los objetos son instancia de una clase específica y heredan sus atributos y las operaciones que estén disponibles para manipular estos atributos. ⁴

Atributos: Usted aprendió que los atributos se vinculan a las clases y que describen a las clases de alguna forma. Un atributo puede tomar un valor definido por un *dominio* enumerado. En la mayoría de los casos, un dominio es simplemente un conjunto de valores específicos. Por ejemplo, la clase automóvil podría tener un atributo color que puede ser cualquiera de los colores predefinidos.⁴

Pressman también destaca que “estas definiciones implican la existencia de una jerarquía de clase en la que los atributos y operaciones de la superclase se heredan por parte de la subclase, que pueden agregar atributos y métodos “privados” adicionales.”

5.1.2. Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual de propósito general que se utiliza para especificar, visualizar, construir y documentar los artefactos de un sistema software. Captura decisiones y conocimiento sobre sistemas que deben ser construidos. Se usa para comprender, diseñar, ojear,

⁴ Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. McGraw Hill.

configurar, mantener y controlar la información sobre tales sistemas. Está pensado para ser utilizado con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre las técnicas de modelado e incorporar las mejores prácticas de software actuales en una aproximación estándar. UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser apoyado por herramientas de modelado visuales e interactivas que dispongan de generadores, tanto de código, como de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos existentes.⁵

UML capta la información sobre la estructura estática y el comportamiento dinámico del sistema. Un sistema es modelado como una colección de objetos discretos que interactúan para realizar un trabajo que en última instancia beneficia a un usuario externo. La estructura estática define tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre los objetos. El comportamiento dinámico define la historia de los objetos a lo largo del tiempo y la comunicación entre objetos para cumplir los objetivos. El modelado de un sistema desde varios puntos de vista separados pero relacionados, permite entenderlo para diferentes propósitos.⁵

5.1.3. Herramienta CASE

Enterprise Architect Corporate Edition es una herramienta CASE para el diseño y construcción de sistemas software. Enterprise Architect Corporate Edition soporta

⁵ James Rumbaugh, G. B. (2007). *El Lenguaje Unificado de Modelado Manual de referencia*. Madrid: Pearson.

la especificación UML 2.0, que describe un lenguaje visual que permite la definición de los modelos de un proyecto.⁶

A pesar de que en el mercado existen muchas soluciones similares a la de Sparx Systems¹, su producto estrella, es decir Enterprise Architect, trae consigo muchas características interesantes que potencializan la generación de modelos de una forma intuitiva y fácil para quien se inicia en el mundo del modelamiento, y que, además, quiere obtener un valor agregado en su trabajo profesional.⁶

5.1.4. Diagramas UML

Según Rumbaugh en su libro UML manual de referencia un diagrama es “una representación gráfica de una colección de elementos del modelo, habitualmente presentado como un gráfico conectado de arcos (relaciones) y vértices (otros elementos del modelo). UML dispone de cierto número de tipos de diagrama.”

5.1.5. Diagrama de casos de uso

Un diagrama de casos de uso es un gráfico de actores, un conjunto de casos de uso englobados por un límite de sujeto (un rectángulo), asociaciones entre los actores y los casos de uso, relaciones entre los casos de uso, y generalización entre los actores. Los diagramas de casos de uso muestran elementos del modelo de casos de uso.⁵

Los casos de uso también se pueden describir a varios niveles de detalle. Se pueden descomponer en partes y ser descritos en términos de otros casos de uso

⁶ Enterprise Architect. (s.f.). *Manual de usuario* . Retrieved from Docplayer : <http://docplayer.es/1170834-Enterprise-architect.html>

más simples. Un caso de uso se implementa como una colaboración en la vista de interacción.⁷

En el nivel del sistema, los casos de uso representan el comportamiento externo del sistema tal y como lo ven los usuarios externos. Sin embargo, a diferencia de una operación, un caso de uso puede continuar recibiendo entradas de sus actores durante su ejecución. Los casos de uso se pueden aplicar al sistema completo, pudiéndose aplicar también internamente a unidades más pequeñas del sistema, como subsistemas y clases individuales. Un caso de uso interno representa el comportamiento que presenta un subsistema al resto del sistema. Por ejemplo, un caso de uso de una clase representa un trozo coherente de funcionalidad que la clase proporciona a otras clases que juegan ciertos roles dentro del sistema. Una clase puede tener más de un caso de uso.⁷

5.1.6. Diagrama de clases

Un diagrama de clases es una presentación gráfica de la vista estática que muestra una colección de elementos declarativos (estáticos) del modelo, como clases, tipos y sus contenidos y relaciones. Un diagrama de clases puede mostrar una vista de un paquete y puede contener símbolos para los paquetes anidados. Un diagrama de clases contiene ciertos elementos de comportamiento, como operaciones, pero sus dinámicas se expresan en otros diagramas, como en los diagramas de estados y los diagramas de comunicación.⁷

5.1.7. Diagrama de comunicación

Diagrama que muestra interacciones organizadas alrededor de las partes de una estructura compuesta o los roles de una colaboración. A diferencia de un diagrama de secuencia, un diagrama de comunicación muestra explícitamente las relaciones

⁷ James Rumbaugh, G. B. (2007). *El Lenguaje Unificado de Modelado Manual de referencia*. Madrid: Pearson.

entre los elementos. Por otro lado, un diagrama de comunicación no representa el tiempo como una dimensión separada, por lo que la secuencia de los mensajes y los hilos concurrentes se deben determinar utilizando números de secuencia. Tanto los diagramas de secuencia como los de colaboración expresan interacciones, pero las muestran de diferente forma.⁸

5.2. Plan Operativo Anual

El plan operativo anual es un documento formal en el que se enumeran, por parte de los responsables de una entidad (compañía, departamento, sucursal u oficina) los objetivos a conseguir durante el presente ejercicio.⁹

El plan operativo anual debe estar perfectamente alineado con el plan estratégico de la empresa, y su especificación sirve para concretar, además de los objetivos a conseguir cada año, la manera de alcanzarlos que debe seguir cada entidad (departamento, sucursal, oficina...).⁹

Según Pete Babich en su libro *Hoshin Handbook* “la mayoría de gente estaría de acuerdo con la afirmación de que un proceso de planeación efectiva es crítico para el éxito a largo plazo de la empresa. Sin embargo, para ser efectiva, la planeación debe estar integrada al proceso de dirección de negocios global de la empresa.”¹⁰

⁸ James Rumbaugh, G. B. (2007). *El Lenguaje Unificado de Modelado Manual de referencia*. Madrid: Pearson.

⁹ Sinnexus. (2010, Agosto 5). Retrieved from http://www.sinnexus.com/business_intelligence/plan_operativo_anual.aspx

¹⁰ Babich, P. (1998). *Hoshin Handbook*. California: Total Quality Engineering Inc.

5.3. Sistemas de información

Un Sistema de Información, es aquél que permite recopilar, administrar y manipular un conjunto de datos que conforman la información necesaria para que los estamentos ejecutivos de una organización puedan realizar una toma de decisiones informadamente. En resumen, es aquel conjunto ordenado de elementos (no necesariamente computacionales) que permiten manipular toda aquella información necesaria para implementar aspectos específicos de la toma de decisiones.¹¹

UCINF también detalla que los Sistemas de Información difieren de cualquier otro software por dos razones principales:

- Almacenan gran cantidad de Información.
- Realizan un bajo grado de procesamiento sobre la información, y éste es, fundamentalmente, de tipo estadístico.

Un sistema de información está compuesto generalmente por seis elementos que son:

- Base de Datos: es donde se almacena toda la información que se requiere para la toma de decisiones. La información se organiza en registros específicos e identificables.
- Transacciones: corresponde a todos los elementos de interfaz que permiten al usuario: consultar, agregar, modificar o eliminar un registro específico de Información.
- Informes: corresponden a todos los elementos de interfaz mediante los cuales el usuario puede obtener uno o más registros y/o información de tipo estadístico (contar, sumar) de acuerdo con criterios de búsqueda y selección definidos.
- Procesos: corresponden a todos aquellos elementos que, de acuerdo con una lógica predefinida, obtienen información de la base de datos y generan nuevos registros de información. Los procesos sólo son controlados por el usuario.

Usuario: identifica a todas las personas que interactúan con el sistema, esto incluye desde el máximo nivel ejecutivo que recibe los informes de estadísticas procesadas, hasta el usuario operativo que se encarga de recolectar e ingresar la información al sistema.¹¹

5.4. Base de datos

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una Biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papeles indexados para su consulta.¹²

Los hechos generales de los sistemas gestores de bases de datos son los de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.¹²

5.5. Gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.¹³

¹¹ UCINF. (2000, 03 23). *Sistemas de información*. Retrieved from <http://jms.caos.cl/si/index.html>

¹² EcuRed. (s.f.). *Bases de datos*. Retrieved from EcuRed: https://www.ecured.cu/Base_de_Datos

Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.¹³

5.6. Software

La plataforma de desarrollo que se pretende utilizar es Open source, o sea de código abierto, una parte de las aplicaciones y sistemas dentro de INETER están desarrolladas en el entorno de programación .Net® de Microsoft® y se quiere probar nuevas tecnologías con auge en el mercado, y se ha establecido previamente por la Dirección de Sistemas Geo informáticos debido a su buena aceptación en el mercado de tecnologías actuales.

El gestor de base de datos que se ocupa es PostgreSQL, ya que es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.¹⁴

¹³ EcuRed. (s.f.). *Sistema Gestor de Base de Datos*. Retrieved from EcuRed: https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos

¹⁴ Interpolados. (2016, 11, 06) *PostgreSQL* Retrieved from <https://interpolados.wordpress.com/2016/11/06/postgresql-2/>

5.6.1. Lenguaje de programación

Se utiliza el lenguaje de programación Ruby®¹⁵, es un lenguaje basado en objetos, además de ser uno de los de mayor crecimiento y aceptación en la comunidad desarrolladora. Su elegante sintaxis se siente natural al leerla y fácil al escribirla.¹⁶

Para el sistema también se utiliza el framework Ruby on Rails® también conocido como RoR, es un framework de desarrollo web escrito en Ruby que sigue el patrón de tres capas Modelo-Vista-Controlador (MVC) y facilita el desarrollo de aplicaciones webs, que está optimizado para la satisfacción de los programadores y para la productividad sostenible. Este está respaldado por más de dos mil colaboradores que contribuyen a que el código sea cada vez más eficiente y versátil. Te permite escribir un buen código evitando la ambigüedad y favoreciendo la convención antes que la configuración.¹⁶

5.6.2. Web socket

Web socket representa una larga espera de evolución en la tecnología web de cliente/servidor. Esto permite mantener una conexión a través de puertos TCP entre el cliente y el servidor, lo cual permite una comunicación bidireccional, permitiendo los mensajes ser instantáneamente distribuidos consumiendo menos recursos [...] Web Socket representa el estándar para comunicaciones bidireccionales en tiempo real entre clientes y servidores, primeramente, en los navegadores web y llegando hasta cualquier cliente o servidor. El primer enfoque del estándar significa que nos encargamos de la funcionalidad que va a trabajar de la misma manera en múltiples plataformas. (Pusher, s.f.)¹⁷

¹⁵ Es un lenguaje de programación dinámico, flexible y completamente orientado a objetos, enfocado en la simplicidad y productividad

¹⁶ Ruby Org. (s.f.). *Ruby*. Extraído de: <https://www.ruby-lang.org/es/>

¹⁷ Pusher. (s.f.). What are WebSockets?. Extraído de: <https://pusher.com/websockets>

5.6.3. Redis

Redis es un almacén de estructura de datos en memoria de código abierto (con licencia de BSD), que se utiliza como base de datos, caché y agente de mensajes. Admite estructuras de datos tales como cadenas, hashes, listas, conjuntos, conjuntos ordenados con consultas de rango, mapas de bits, hiperlogálogos e índices geoespaciales con consultas radiales. Redis tiene una replicación incorporada, secuencias de comandos Lua, desalojos LRU, transacciones y diferentes niveles de persistencia en disco, y proporciona alta disponibilidad a través de Redis Sentinel y particiones automáticas con Redis Cluster.¹⁸

5.7. Servidor web

Servidor Web es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.¹⁹

En este caso particular se utiliza un servidor llamado Phusion Passenger un servidor web y de aplicaciones, diseñado para ser rápido, robusto y liviano, que soporta aplicaciones de Ruby, Python, node.js y meteor.²⁰

¹⁸ Redis. (s.f.). Introducción a redis. Extraído de: <https://redis.io/topics/introduction>

¹⁹ EcuRed. (s.f.). *Servidor Web*. Extraído de: https://www.ecured.cu/Servidor_Web

²⁰ Phusion Passenger. (s.f.). github. Extraído de: <https://github.com/phusion/passenger>

5.8. Aplicación web

En la Ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación (Software) que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.²¹

EcuRed también define “Las aplicaciones web son populares debido a lo práctico del navegador web como Cliente ligero, a la independencia del Sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.”

Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.²²

Según EcuRed, algunas ventajas de las aplicaciones web:

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.

²¹ EcuRed. (s.f.). *Aplicacion web*. Extraído de: https://www.ecured.cu/Aplicaci%C3%B3n_web

- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.

5.9. Modelo en espiral

El modelo en espiral, propuesto originalmente por Boehm, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas.²²

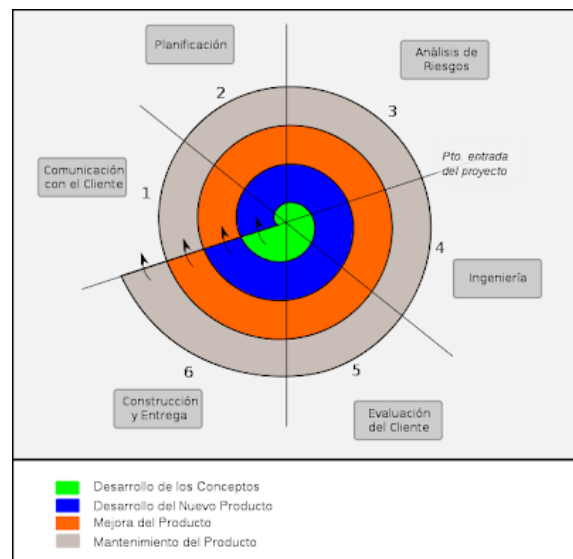


Figura 2 Modelo en espiral

²² Grijalva, N. (15 de 10 de 2012). *Ingeniería de software 1*. Obtenido de <http://software1nathalygrijalva.blogspot.com/2012/10/modelo-espiral.html>

El general los beneficios de usar modelo en espiral pueden ser los siguientes:

- Se pueden gestionar las expectativas de cliente (requisitos desarrollados, velocidad de desarrollo, calidad) de manera regular, puede tomar decisiones en cada iteración.
- La finalización de cada iteración es el lugar donde el equipo puede decidir cómo mejorar su proceso de trabajo, en función a la experiencia obtenida.
- Permite mitigar desde el inicio riesgos del proyecto. ²²

5.10. Técnicas de recopilación de información

Las técnicas de recolección de información son procedimientos especiales utilizados para obtener y evaluar las evidencias necesarias, suficientes y competentes que le permitan formar un juicio profesional y objetivo, que facilite la calificación de los hallazgos detectados en la materia examinada. (Batista, 2010)

Las técnicas verbales pueden ser: entrevistas, encuestas, y cuestionarios.

Oculares: verificar en forma directa y paralela, la manera como los responsables desarrollan y documentan los procesos o procedimientos, mediante los cuales la entidad evaluada ejecuta las actividades objeto de evaluación. (Batista, 2010)

Documentales: Obtener información escrita, para soportar las afirmaciones, análisis o estudios realizados por los actores. Estas pueden ser: comprobación; y revisión analítica. (Batista, 2010)

6. Desarrollo del sistema

El desarrollo del sistema hoja de tiempo se realizó siguiendo la metodología de espiral, con el fin de facilitar la comunicación entre los directivos y usuarios del sistema, para cumplir los objetivos se definieron 5 fases ver *figura 2*, obteniendo retroalimentación y un prototipo de cada iteración, reduciendo riesgos y haciendo el proceso de desarrollo rápido de versiones incrementales desde las versiones de prototipo y diseño hasta las versiones cada vez más completas del sistema diseñado.

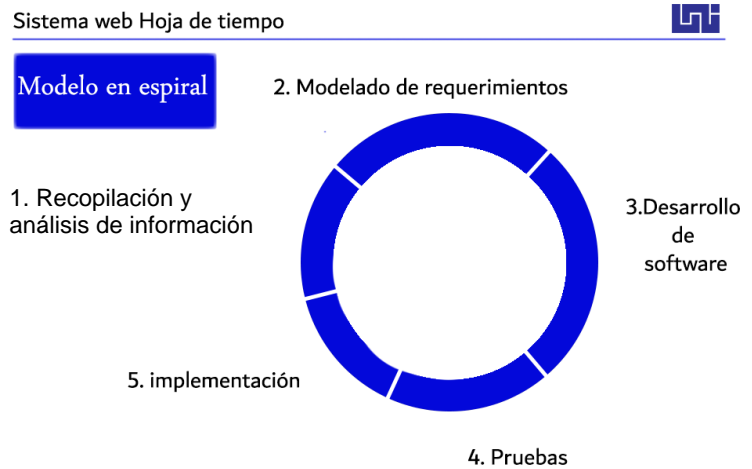


Figura 3 Espiral sistema hoja de tiempo

La primera fase del análisis de la información se reúnen todos los requerimientos en base a técnicas de recopilación anteriormente definidas, se realizarán entrevistas con varios usuarios para definir el dominio del problema, así como reuniones para obtener la mayor cantidad de información y definir los requerimientos funcionales y principales características del sistema.

La fase de modelado de requerimientos es donde se definen mediante diagramas y artefactos de ingeniería de software todas las funcionalidades, así como interfaces, estilos y riesgos que se pueden prever en el desarrollo del sistema, aquí también se definen alcances y visiones generales se trata de entender el dominio del problema.

La fase de desarrollo de software se codifica los módulos en base a los requisitos y funciones que estén especificadas en la fase anterior, dejando listas las funcionalidades para la siguiente fase.

Fase de pruebas, en esta etapa se necesita probar cada módulo que se haya creado o modificado para confirmar que el sistema está en las mejores condiciones para ser usado por los usuarios finales.

La última fase es la implementación donde se publica al servidor para ser usado por el personal de la institución. Y monitorear que el sistema cumple con las expectativas de funcionalidad y objetivos propuestos.

En cada iteración vamos obteniendo una nueva versión del software o una nueva funcionalidad prevista, con este modelo se pretende evolucionar el producto haciendo una entrega incremental a partir de requerimientos obtenidos en las fases anteriores, añadiendo nuevos objetivos/requisitos o mejorando los que ya fueron completados.

6.1. Recopilación y análisis de la información

Como métodos de recolección de información se definieron reuniones con los encargados de la Dirección de Sistemas Geo Informáticos del instituto, donde hay 3 subdirecciones: IT, Desarrollo, Geo repositorio (Base de datos), y entrevistas para definir el producto. *Ver anexo 1 y 2.*

En esta fase se definen los aspectos generales del sistema sus funcionalidades para cada iteración, se definen los procesos de planificación para la generación del plan operativo, así como estrategias para el auto seguimiento de las actividades, automatización de la comunicación y los reportes que se necesitan para definir las variables y modelos.

6.1.1. Requerimientos generales

A continuación, se enumeran los requerimientos funcionales y no funcionales obtenidos en base a las entrevistas y reuniones realizadas, para obtener la mayor

cantidad de información posible para definir el dominio del problema, las cuales posteriormente serán analizadas si es posible implementar en el sistema.

6.1.2. Reglas del negocio

Las siguientes reglas se definieron en conjunto con el equipo de trabajo de la dirección general de sistemas geo-informático, para entender un poco mejor el dominio del problema y diseñar una solución que se adapte mejor a los procesos de la institución.

- El director y subdirector pueden ver el trabajo ingresado por los empleados que corresponda con el área, y estos serán los encargados de validar los avances ingresados por los usuarios.
- Los usuarios empleados deberán llenar su trabajo para ser validado.
- Los directores y subdirectores deben dar seguimiento a los proyectos como el total de avances realizados, el porcentaje de avance de cada proyecto, el tiempo invertido en proyectos, a las tareas, el tiempo por tarea, así como también la información de los usuarios.
- Los subdirectores deben validar el avance diario de sus empleados a cargo, los directores podrán validar los avances de los subdirectores a cargo.
- El sistema debe funcionar sin acceso a internet sin ningún problema, esto ya que en algunas direcciones no hay acceso a internet por motivos de seguridad.
- El sistema debe implementar canales de comunicación como un canal de chat en vivo o foros que permita mejorar la comunicación dentro del instituto.

6.1.3. Requerimientos específicos

6.1.3.1. *Requerimientos funcionales*

Las siguientes son funciones que el sistema permite según su rol:

- **Administrador:**

- Administrar catálogos de posiciones, categorías, locales, administrar usuarios del sistema.

- **Director:**

- Gestionar categorías para proyectos
- Gestionar proyectos, actividades y tareas, ya sea normal o repetitivo.
- Incrementar porcentaje de avance de tareas según se vayan trabajando
- Validar los avances de los usuarios subdirectores o empleados que estén por validar
- Agendar reuniones
- Puede ver la información relacionada con su personal a cargo ya sea subdirectores o empleados, puede ver informes gráficos con información de su personal a cargo. Seguimiento de tareas, ver avances de actividades o tareas, así como los tiempos
- Enviar mensajes a los demás directores, subdirectores, y empleados

- **Subdirector:**

- Crear categorías para proyectos
- Puede gestionar proyectos, actividades y tareas, de la misma manera
- Cambiar tareas de estado e incrementar porcentaje de avance de tareas según se vayan trabajando y completando
- Validar los avances de los empleados que estén por validar
- Agendar reuniones
- Puede ver la información relacionada con su personal a cargo ya sea subdirectores o empleados, puede ver informes gráficos con

información de su personal a cargo. Seguimiento de tareas, ver avances de actividades o tareas, así como los tiempos usados

- Enviar mensajes al director de área, subdirectores de su área y a personal a cargo

- **Empleado:**

- Enviar mensajes al director de área, subdirector de su área y empleados de su área
- Gestionar tareas
- Registrar avances de actividades diarias
- Comentar reuniones

6.1.3.2. Requerimientos no funcionales

A continuación, algunos requerimientos no funcionales más importantes que la institución desea para el sistema:

- La estructura organizacional de los roles por usuario será entregada por el departamento de RRHH *ver figura 1*.
- La base de datos deberá ser registrada con un usuario para la aplicación, y también auditada con una tabla la cual se actualiza cuando hay algún cambio en ciertas tablas previamente establecidas mediante disparadores.
- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 2 días, aunque se definió una semana de inducción por cada dirección para atender a todo el personal de cada área.
- El sistema debe contar con manuales de usuario estructurados adecuadamente.
- El sistema debe proporcionar mensajes de error y de información concisos.
- El sistema debe poseer un diseño responsivo para ser accedido desde tabletas, celulares y pc.

- El sistema debe tener gráficas bien formadas, y formularios de consulta para extraer información necesaria.
- Se deben implementar herramientas que agilice la comunicación de los empleados ya sean foros o un chat en vivo.

6.1.4. Estudio de factibilidad

En esta sección se determinan los diferentes aspectos que verifican la viabilidad del sistema, se detallan algunos aspectos más importantes tales como la factibilidad técnica que contempla las necesidades tanto para el software como para el desarrollo e implementación; la factibilidad operativa que se enfoca en las reglas de procesos del sistema para que este pueda ser usado de la mejor manera; la factibilidad económica que contempla los costos de desarrollo, equipos y lo necesario para que el sistema sea puesto en marcha; finalmente la factibilidad legal que define el marco de legalidad que rige la aplicación y el uso de esta.

6.1.4.1. *Factibilidad técnica*

En esta sección se verifica que tan factible es la implementación del sistema en el aspecto que aborda los requisitos mínimos de hardware y software necesarios para el desarrollo y uso de la herramienta.

✓ Requisitos mínimos del sistema

Para obtener un correcto funcionamiento del sistema y sacar mayor provecho de su utilización se recomiendan las siguientes características para los equipos destinados, siempre dejando libre la opción de mejorar estas especificaciones:

a) Servidor web:

- Para el servidor web: Este puede ser un servidor físico o máquina virtual dentro del centro de datos de la institución un lugar seguro para este y contar al menos con las siguientes características:
 - El sistema operativo del servidor debe ser un Debian® versión 8 Jessie® con un procesador Intel al menos de 3.30GHz (con dos núcleos) con al menos 3GB de memoria RAM.

b) Equipo de desarrollo

- Para el desarrollo: Un equipo con sistema operativo Ubuntu 16.04, procesador Intel I3 o superior, de al menos 2.4GHz, con al menos 4GB de memoria RAM, herramientas open source que se encuentran en la web como el editor de código Atom, Postgres 9.6, Ruby, Rails y redis que se encuentran gratuitamente en sus sitios oficiales y auxiliándonos del navegador web Google Chrome para el desarrollo.

c) Software de usuario

Para poder acceder al sistema sin inconvenientes es necesario que los usuarios tengan:

- Navegador actualizado ya sea Chrome o Firefox.
- Conexión a la red interna del instituto

✓ Verificación de la factibilidad técnica para el sistema

Se procede a la verificación de cumplimiento de requisitos tanto hardware como software, obteniendo las siguientes observaciones:

a) Hardware:

- Los equipos con los que se cuenta en el instituto suplen las necesidades de requerimientos mínimos para el sistema.
 - El servidor cuenta con un Intel® Xeon® CPU E5-2643 de 3.30GHz (con dos núcleos) y 4GB de memoria RAM. Con un sistema operativo Debian ya especificado

- El servidor de base de datos es un Windows server 2016 con 4 núcleos, 16Gb en RAM, cuenta con dos discos flash de 140Gb en total y Postgresql 9.6
- El equipo de desarrollo es un computador Lenovo propio del desarrollador con un procesador Intel Core I5-3320M @ 2.6GHz de 4 núcleos, con 8GB de memoria RAM

b) Software de desarrollador:

- Todos los elementos de software que se usaron para el desarrollo son de licencia de software libre desde las herramientas de desarrollo, pasando por la base de datos y el servidor web el cual es basado en nginx. O sea que el costo de desarrollo es principalmente el personal.

c) Software de usuarios:

- El software ya lo tienen todos los equipos que van a ser utilizados este puede ser cualquier navegador ya sea Chrome o Firefox.

6.1.4.2. Factibilidad operativa

Para obtener el mejor provecho de la aplicación se pretende que sea lo más amigable posible, intuitiva y fácil de usar para que el usuario pueda adaptarse fácilmente y aprovechar completamente todas las herramientas que el sistema posee, ahorrando tiempo en estos procesos.

Una vez creada la máquina virtual donde será alojado la aplicación procedemos a instalar todo el software que se necesita para poder alojar la aplicación con todos sus recursos, se configura el servidor Passenger para escuchar al puerto 443 y servir las peticiones de los usuarios del sistema, comunicarse con la base de datos y también se configura en el mismo un servidor de Redis el cual va a escuchar al puerto 6379 para servir como memoria caché para la aplicación de chat.

El sistema estará disponible para toda la red del instituto en la dirección <https://hojadetiempo.ineter.gob.ni>. La autenticación de usuario es mediante la aplicación y los roles mediante el registro de recursos humanos de la base de datos SIRH y utilizando cualquier navegador el usuario podrá acceder a su información de manera dinámica y facilitando el auto seguimiento a sus actividades.

Se han usado librerías de front end como MaterializeCss, JQueryUI, moment.js, go.js y Highchart.js con el fin de obtener la mejor experiencia de usuario y hacer de la aplicación mucho más fácil de usar y entender por cualquier tipo de usuario ya sea un usuario experto o un usuario inexperto. Además de adaptarnos al diseño minimalista de Material Design.

La aplicación se comunica con el servidor de bases de datos haciendo consultas a dos bases de datos que almacenan la información una operativa y la otra de recursos humanos. Se muestran diferentes herramientas como calendarios con información de los avances de tareas y reuniones, reportes de tiempo y tareas mediante gráficos, aplicación de mensajería en tiempo real, las cuales se han sido agregados en diferentes ciclos de desarrollo dependiendo de las necesidades que van surgiendo, cabe mencionar que los gráficos dinámicos están sujetos a cambios por parte del instituto

El servidor de base de datos se aloja dentro de la misma red en otro servidor especial Windows server 2016 en este servidor actualmente se usa para almacenar información de varios sistemas del instituto.

6.1.4.3. Factibilidad económica

El instituto cuenta con el hardware y el software necesario para llevar a cabo este proyecto, así como la infraestructura donde el software va a ser probado y desplegado una vez se cuente con avance significativo de cada iteración, el instituto ha sido parte importante de este proceso monográfico aportando todos los recursos tecnológicos y necesarios para la realización de esta monografía.

Con efectos investigativos se ha incluido el costo de llevar este proyecto a cabo mediante la compra del alojamiento en GoDaddy un portal de hosting de paga para servidores y aplicaciones con una proyección de un año para el desarrollo del sistema el 2do año puede variar el precio, aunque el instituto cuenta con la infraestructura para desplegar el sistema dejando en claro que el costo de alojamiento no ha sido costado por la institución.

El instituto ha asegurado el pago del desarrollo del sistema y se ha sido desglosado con el objetivo meramente informativo para fines de esta monografía en la siguiente tabla de costos totales.

El total el costo de implementar este software con un desarrollador que se encargue completamente del software y el servidor para implementación sería de un poco más de cuatro mil dólares durante un año, los cuales pueden variar dependiendo del host o si se contrata otra persona para este proyecto.

Tabla 1 Costos de desarrollo

#	Descripción	Cant.	Precio	Total
1	Software con licencias open source: Ruby, Rails, Postgresql, Redis, Debian, etc.	6	\$ 0	\$ 0
2	Desarrollador Ruby on Rails x 12 meses	12	\$ 350	\$ 4,200
3	Analista de sistemas x3 meses	3	\$ 450	\$ 1,350
3	Hosting económico en GoDaddy x 12 meses	12	\$ 8	\$ 96
			Total	\$ 5,646

Las herramientas open source cuentan con licencias de uso gratuitas y descargas oficiales de las herramientas directamente de sus sitios oficiales, haciendo el desarrollo completamente viable y factible.

6.1.4.4. *Factibilidad legal*

El presente trabajo cuenta con los permisos legales necesarios y cumple con la ley de derechos de autor para software libre respetando todos los acápite que dicha ley establece. INETER cuenta con un marco legal el cual establece que la dirección de Sistemas Geo informáticos puede destinar el uso de recursos necesarios y personal para el desarrollo de herramientas respetando el derecho de propiedad intelectual del desarrollador y poniendo a disposición la aplicación para futuras implementaciones en instituciones del gobierno que sea debidamente solicitado y autorizado por la institución.

6.2. Modelado de requerimientos

En esta fase se realiza el análisis y diseño de la aplicación, se obtiene la arquitectura de la misma, funcionalidades mediante la implementación de modelos de UML para los distintos enfoques de la aplicación. El modelado de los datos mediante diagramas de objetos, la infraestructura se define en diagrama de despliegue, las actividades mediante diagramas de secuencia.

6.2.1. Modelado de casos de uso

Se presentan los diagramas de caso de uso principales y secundarios del sistema, modelados durante la primera iteración. Junto con su descripción de caso de uso en la que se describe en qué consiste cada caso de uso representado en el diagrama, los casos de uso se clasificaron según los procesos de la institución obtenidos durante entrevistas con el personal.

6.2.1.1. Caso de uso de inicio de sesión

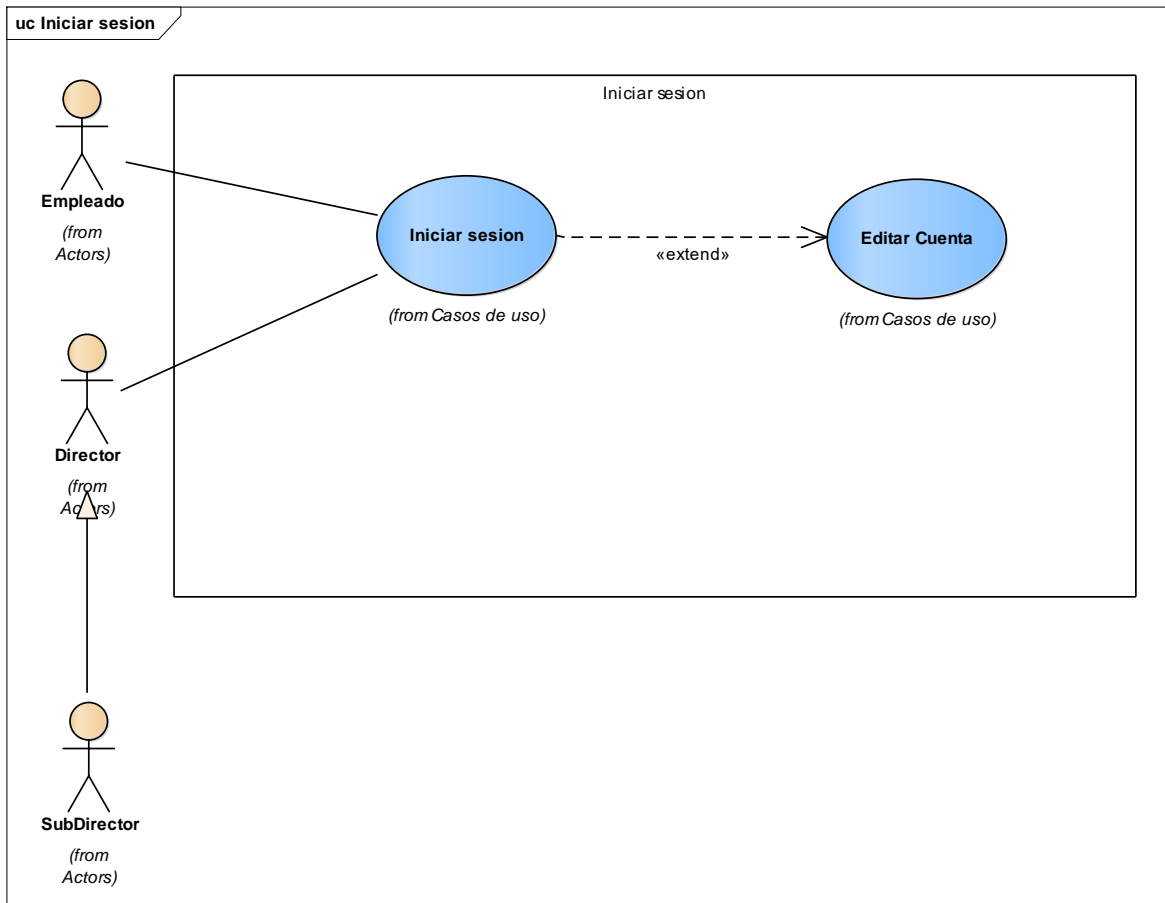


Figura 4 Diagrama de inicio de sesión

- Descripción de caso de uso de inicio de sesión

Caso de uso: 1	Iniciar sesión
Actores	Director, Sub director, Empleado
Objetivos	Autenticar el usuario para su ingreso al sistema
Precondiciones	Debe haber registrado y creado una cuenta de usuario.
Postcondiciones	Una vez iniciada la sesión el usuario puede hacer uso del sistema, que se refleja en los diagramas siguientes.
Manejo de situaciones excepcionales	Si no puede acceder al sistema por olvido de sus credenciales, podrá restablecer su contraseña usando su correo de registro.

Flujo de eventos	1	<ol style="list-style-type: none"> 1. El usuario debe ingresar a la aplicación. 2. El sistema mostrará una vista de inicio de sesión, donde se ingresan las credenciales: usuario y contraseña. 3. Si las credenciales son válidas podrá acceder al sistema.
------------------	---	---

6.2.1.2. Caso de uso de recuperación de contraseña

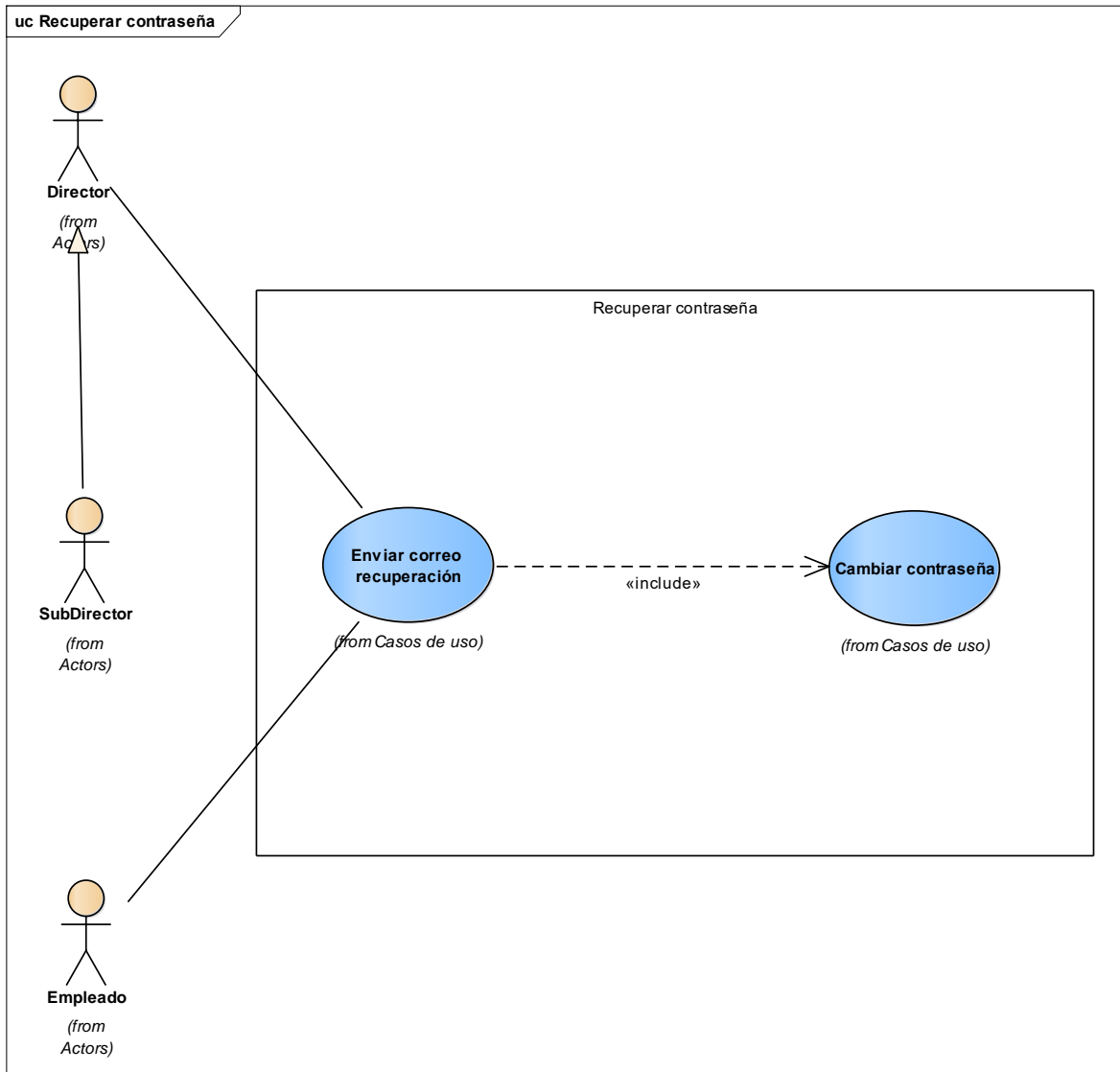


Figura 5 Caso de uso recuperación de contraseña

- Descripción de caso de uso recuperación de contraseña

Caso de uso: 2	Recuperar contraseña	
Actores	Director, Sub director, Empleado	
Objetivos	Reestablecer la contraseña de una cuenta de usuario	
Precondiciones	La cuenta de usuario debe existir	
Pos condiciones	Se enviará un enlace al correo de registro para reestablecer la contraseña.	
Manejo de situaciones excepcionales	Para reestablecer la contraseña el usuario tiene un plazo de 24 horas, sino lo restablece en ese tiempo el enlace se invalidará.	
Flujo de eventos	1	1. Clic en la opción ¿Olvidaste tu contraseña? 2. Escribir el correo de registro de usuario y dar clic en “Enviar” 3. Revisar la cuenta de correo y abrir el enlace para reestablecer la contraseña 4. Escribir nueva contraseña y aceptar 5. Ingresar al sistema

6.2.1.3. Caso de uso registrar usuario

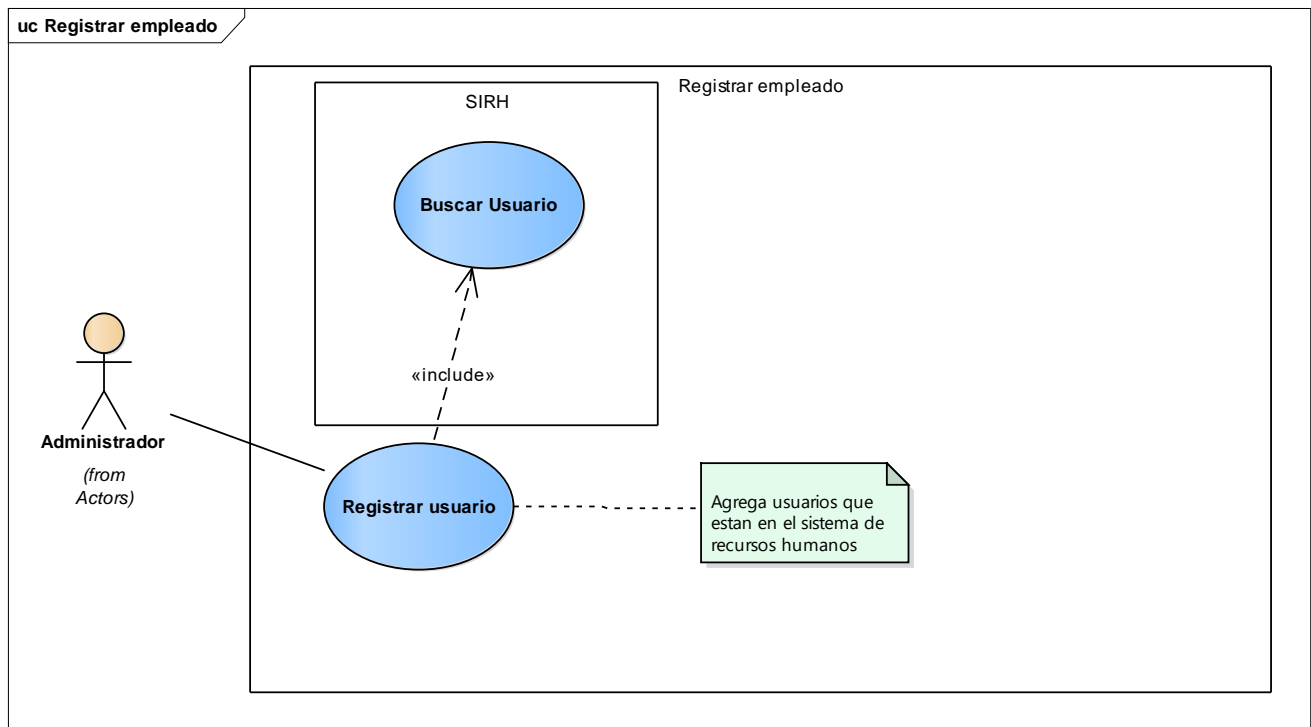


Figura 6 Caso de uso registrar empleado

- Descripción de caso de uso registrar usuario

Caso de uso: 3	Registrar usuario	
Actores	Administrador	
Objetivos	Ingresar un nuevo usuario al sistema	
Precondiciones	La cuenta de usuario debe existir en el sistema de recursos humanos	
Pos condiciones	Una vez registrado el usuario puede ser deshabilitado	
Manejo de situaciones excepcionales	Para reestablecer la contraseña el usuario tiene un plazo de 24 horas, sino lo restablece en ese tiempo el enlace se invalidará.	
Flujo de eventos	1	1. Clic en la opción Usuarios en la ventana de configuración

		2. Clic en el botón nuevo usuario 3. Seleccionar un usuario a agregar 4. Enviar
--	--	---

6.2.1.4. Caso de uso general del sistema



Figura 7 Caso de uso general

- Descripción de caso de uso general del sistema

Caso de uso: 4	Sistema hoja de tiempo	
Actores	Director, Sub director, Empleado	
Objetivos	Gestionar los proyectos, actividades, tareas, reuniones y personal, para mejorar la planificación operativa de las direcciones y sub-direcciones de la institución	
Precondiciones	El sistema será accesible desde la red interna de la institución Se debe de iniciar sesión para acceder al sistema	
Flujo de eventos	1	<p>Gestionar proyectos</p> <p>Precondición 1.1: Para poder agregar un usuario a un proyecto este debe pertenecer a la dirección o sub-dirección correspondiente.</p> <p>Precondición 2.1: Para poder gestionar proyectos se debe tener una estructura de dirección y sub-dirección creada.</p> <p>Sub-flujo:</p> <p>1.1 El director y sub director podrá agregar o eliminar usuarios al proyecto.</p> <p>1.2 El director y sub director podrá gestionar los proyectos a los que pertenece o los que haya creado.</p> <p>1.3 El director y sub director podrá gestionar proyectos especiales de igual manera.</p>
	2	<p>Gestionar actividades</p> <p>Precondición 2.1: Para agregar un usuario a una actividad, este debe haber sido agregado al proyecto.</p> <p>Sub-flujo:</p>

		<p>2.1 El director y sub director podrá gestionar las actividades y los usuarios que forman parte de esta.</p> <p>2.2 El director y sub director podrá manipular las actividades especiales de igual manera.</p>
	3	<p>Gestionar tareas</p> <p>Precondición 3.1: Para agregar usuario a una tarea, este debe pertenecer a la actividad en la que se agrupa la tarea.</p> <p>Sub-flujo:</p> <p>3.1 El director y sub director podrá gestionar sus tareas y los usuarios que forman parte de esta.</p> <p>3.2 El director también manipula las tareas especiales de la misma manera.</p>
	4	<p>Gestionar historial de tarea</p> <p>Precondición: 4.1: Para poder agregar un avance de tarea se debe pertenecer a dicha tarea, así como a su actividad y proyecto padre.</p> <p>Sub-flujo:</p> <p>4.1 Rellenar los datos solicitados, seleccionar la tarea a la cual se registrar el avance</p> <p>4.2 Se puede editar y eliminar los avances de tarea</p>
	5	<p>Gestionar reuniones</p> <p>Sub-flujo:</p> <p>5.1 Los usuarios director y sub-director podrán crear reuniones, ingresando los datos: fecha, hora inicio, hora fin y agregar a los participantes de la misma.</p>

		Pos condición 5.1: Los usuarios que sean agregados a una reunión podrán ver un avance con los datos de la reunión.
6	Ver informes	<p>Sub-flujo:</p> <p>2.1 Los usuarios directores y sub-directores podrán acceder a informes gráficos con información referente a avances de tareas y actividades.</p>

6.2.1.5. Caso de uso para gestionar proyectos

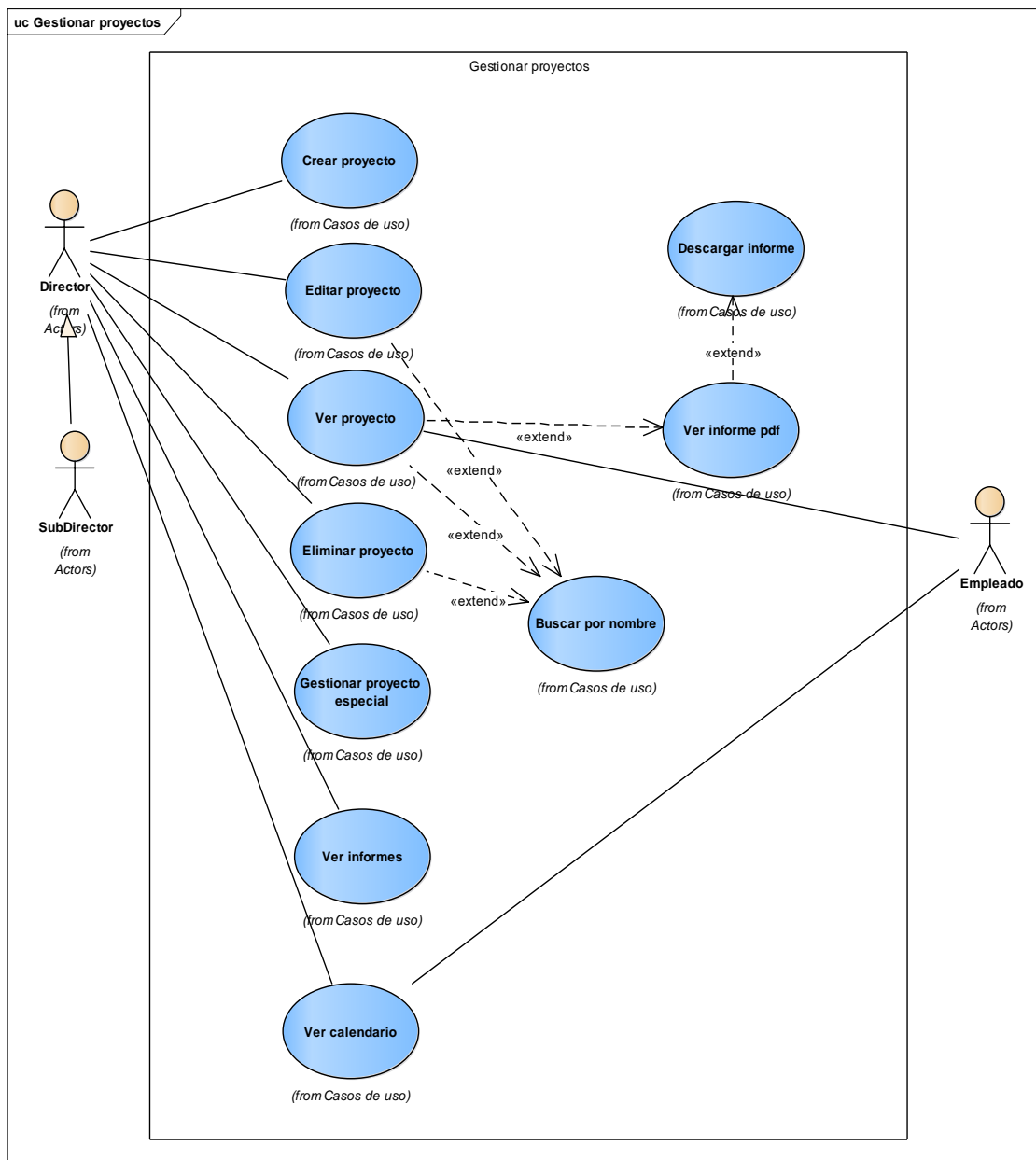


Figura 8 Caso de uso proyectos

- Descripción de caso de uso de proyecto

Caso de uso: 5	Gestionar proyectos	
Actores	Director, Sub director, Empleado	
Objetivos	Gestionar la información de los proyectos de la dirección	
Flujo de eventos	1	<p>Crear proyecto</p> <p>Precondición 1.1: Se deben crear categorías para proyectos</p> <p>Sub-flujo:</p> <p>1.1 Los directores y sub-directores pueden crear proyectos haciendo clic en nuevo proyecto y luego llenar los datos correspondientes.</p> <p>pos condición: Una vez creado el proyecto se pueden crear actividades y agregar usuarios dentro del proyecto.</p>
	2	<p>Editar proyecto</p> <p>Precondición 2.1: Los usuarios permitidos para editar proyectos son directores y sub-directores, los proyectos que pueden editar son los que tengan asociados.</p> <p>Sub-flujo:</p> <p>2.1 Los directores y sub-directores pueden editar proyectos que tengan asignados o hayan creado</p>
	3	<p>Ver proyecto</p> <p>Sub-flujo:</p> <p>2.1 Los usuarios podrán ver los proyectos a los que pertenecen ya sea navegando hacia estos en el panel de navegación del lado izquierdo o ya sea directamente dando clic en el proyecto en la lista de proyectos.</p>

	4	<p>Eliminar proyecto</p> <p>Precondición 4.1: Los proyectos pueden ser eliminados por los directores y sub-directores que formen parte del mismo</p> <p>Sub-flujo:</p> <p>4.1 Se puede buscar un proyecto por nombre</p> <p>4.2 Clic en el icono de eliminar</p> <p>pos condición: Proyecto eliminado</p>
	5	<p>Ver informe</p> <p>Sub-flujo:</p> <p>5.1 Navegar hasta un proyecto</p> <p>5.2 Clic en pdf</p>

6.2.1.6. Caso de uso para gestionar actividades

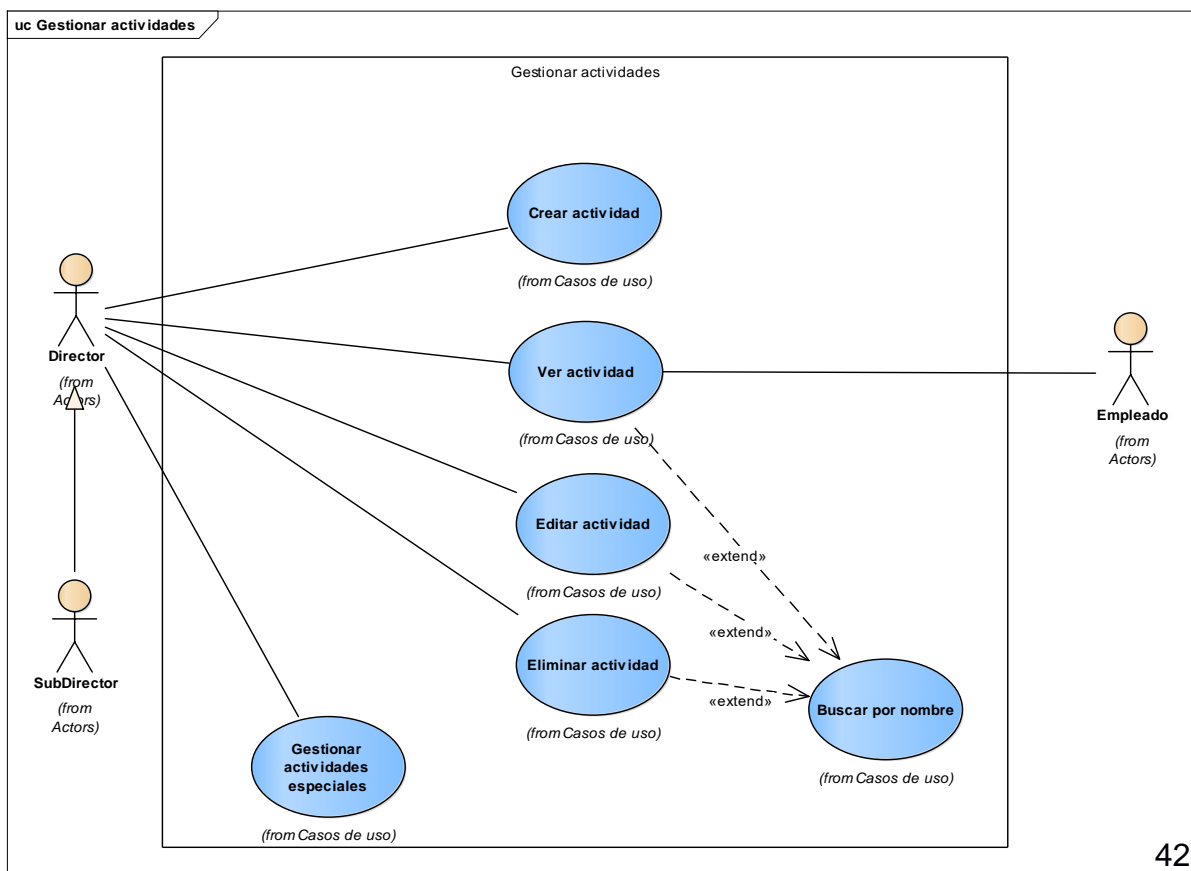


Figura 9 Caso de uso actividades

- Descripción de caso de uso gestionar actividades

Caso de uso: 6	Gestionar actividades	
Actores	Director, Sub director, Empleado	
Objetivos	Gestionar recursos de actividad en el sistema	
Precondiciones	Debe existir un proyecto.	
Flujo de eventos	1	<p>Crear actividad</p> <p>Sub-flujo:</p> <p>1.1 Navegar hasta un proyecto, clic en nueva actividad</p> <p>1.2 Llenar datos: nombre, fecha inicio, fecha fin, prioridad, estado, observación.</p> <p>1.3 Clic en guardar</p>
	2	<p>Editar actividad</p> <p>Precondición 2.1: El usuario debe pertenecer a la actividad, solo los usuarios director y sub-director podrán editar actividades</p> <p>Sub-flujo:</p> <p>2.1 Navegar hasta una actividad o buscar una actividad por nombre</p> <p>2.2 Clic en editar actividad</p>
	3	<p>Ver actividad</p> <p>Sub-flujo:</p> <p>2.1 Seleccionar actividad en lista de actividades o buscar actividad por nombre</p>
	4	<p>Eliminar actividad</p> <p>Precondición 4.1: Los usuarios directores y sub-directores podrán eliminar las actividades solo si pertenecen a estas.</p> <p>Sub-flujo:</p> <p>4.1 Buscar una actividad por nombre</p>

6.2.1.7. Caso de uso para gestionar tareas

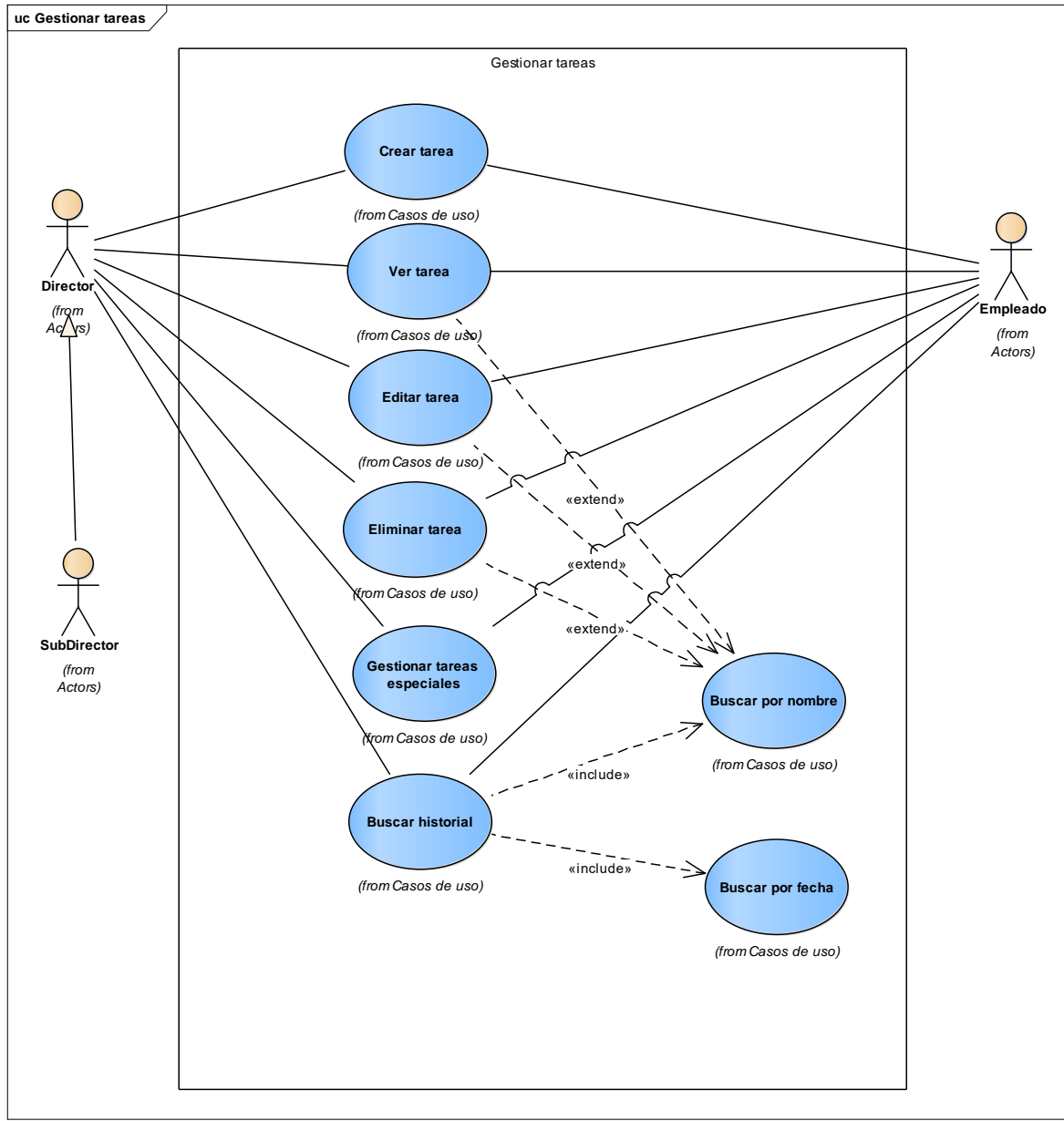


Figura 10 Caso de uso tareas

- Descripción de caso de uso para gestionar tareas

Caso de uso: 7

Gestionar tareas

Actores	Director, Sub director, Empleado	
Objetivos	Gestionar recursos de tareas en el sistema	
Precondiciones	Debe existir una actividad.	
Flujo de eventos	1	<p>Crear tarea</p> <p>Sub-flujo:</p> <p>1.4 Navegar hasta una actividad, clic en nueva tarea</p> <p>1.5 Llenar datos: nombre, fecha inicio, fecha fin, prioridad, estado, observación.</p> <p>1.6 Clic en guardar</p>
	2	<p>Editar tarea</p> <p>Precondición 2.1: El usuario debe pertenecer a la tarea</p> <p>Sub-flujo:</p> <p>2.3 Navegar hasta una tarea o buscar una tarea por nombre</p> <p>2.4 Clic en editar tarea</p>
	3	<p>Ver tarea</p> <p>Precondición 3.1: El usuario debe pertenecer a la tarea</p> <p>Sub-flujo:</p> <p>2.2 Seleccionar tarea en lista de tarea, seleccionar tarea en un actividad o buscar tarea por nombre</p>
	4	<p>Eliminar tarea</p> <p>Precondición 4.1: El usuario debe pertenecer a la tarea</p> <p>Sub-flujo:</p> <p>4.3 Buscar una tarea por nombre</p> <p>4.4 Clic en ícono de bote de basura</p>
	5	<p>Buscar historial</p>

	<p>Precondición 5.1: Los usuarios deben haber llenado su respectivo historial de tarea. Los usuarios directores pueden buscar en historial del personal de la dirección. Los sub-directores pueden buscar en el historial del personal de la sub-dirección. Los empleados pueden ver su historial.</p> <p>Sub-flujo:</p> <p>5.1 Clic en avance por usuario</p> <p>5.2 Elegir un usuario</p> <p>5.3 Elegir una fecha inicio y fecha de fin</p> <p>5.4 Clic en botón enviar</p> <p>Postcondición: Lista de avances realizados con sus datos.</p>
--	--

6.2.1.8. Caso de uso gestionar historial de tareas

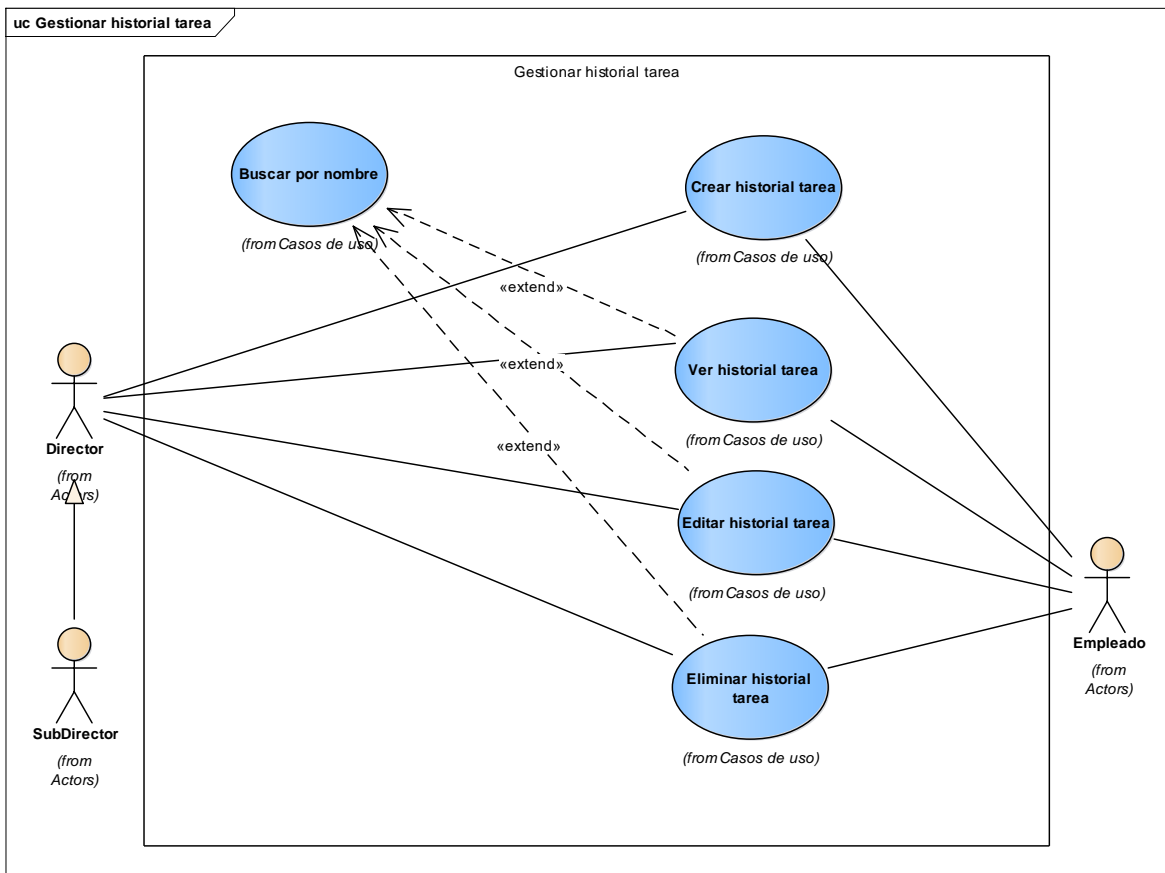


Figura 11 Caso de uso historial de tareas

- Descripción de caso de uso gestionar historial de tareas

Caso de uso: 8	Gestionar historial de tareas	
Actores	Director, Sub director, Empleado	
Objetivos	Gestionar historial de tarea en el sistema	
Precondiciones	Debe existir una estructura de proyecto, actividad y tarea relacionados, para la cual el usuario debe estar asignado a esta.	
Flujo de eventos	1	<p>Crear historial tarea</p> <p>Sub-flujo:</p> <p>1.1 Clic en área de trabajo o navegar hasta el nuevo avance de tarea</p> <p>1.2 Llenar datos: hora inicio, hora fin, fecha, observación y elegir una tarea a la que se va a hacer referencia el nuevo historial</p>
	2	<p>Editar historial tarea</p> <p>Precondición 2.1: El usuario puede editar sus avances</p> <p>Sub-flujo:</p> <p>2.5 Navegar hasta una avance o buscar un avance por descripción</p> <p>2.6 Clic en editar o ícono de lápiz</p>
	3	<p>Ver historial tarea</p> <p>Sub-flujo:</p> <p>2.3 Seleccionar el avance en lista de avances de tarea o buscar avances de tarea por descripción</p>
	4	<p>Eliminar historial tarea</p> <p>Precondición 4.1: Cada usuario puede eliminar sus avances</p> <p>Sub-flujo:</p>

6.2.1.9. Caso de uso para ver informes

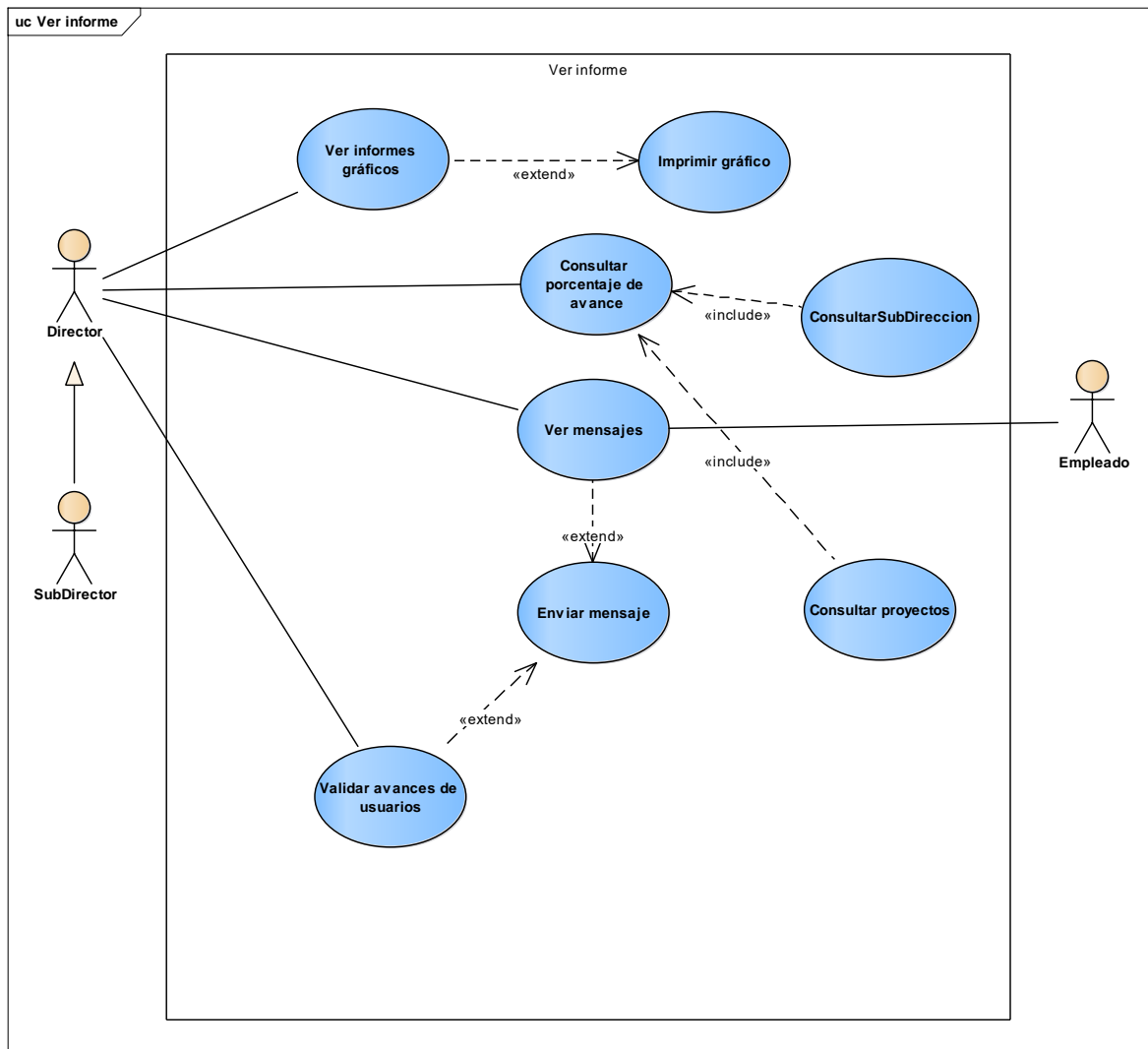


Figura 12 Caso de uso de informes

- Descripción de caso de uso para generar informes

Caso de uso: 9	Ver informe
Actores	Director, Sub director, Empleado
Objetivos	Visualizar la información de los usuarios y sus avances dentro de las actividades

Precondiciones	Los usuarios deben haber llenado su historial de tareas.	
Flujo de eventos	1	Ver informes gráficos

6.2.2. Diagrama de clase

El diagrama de clase es el diagrama principal para el diseño de la aplicación este representa las clases dentro del sistema con sus relaciones, las clases representan sus atributos y algunas operaciones.

Se realiza en base a las especificaciones obtenidas de entrevistas con los expertos del dominio del problema, se basa en las entidades que interactúan dentro de la aplicación y se pudieron identificar las siguientes clases

Categoría: Representa las categorías de los proyectos para ser clasificados.

Proyecto, Actividad, Tarea: Representa un proyecto, actividad o tarea, esta forma la estructura de trabajo para cualquier tipo de proyecto ya sea especial o normal. La diferencia de estos es que uno tiene fin y el otro no.

Avance de tarea: Representa el historial de avances que realizan los usuarios dentro de un lapso en el que se hace referencia a una tarea y una descripción de lo que se ha realizado.

Rol: Como se desarrolla el usuario dentro del recurso (proyecto, actividad o tarea), Desde la versión 2.0.0 tu rol es tu cargo dentro del sistema de recursos humanos.

Usuario: El usuario este puede ser de cualquier rol.

Empleado: Registro de usuario de la institución.

Dirección, Subdirección, Área subordinada, Oficina, Puesto: Información del nivel del trabajador dentro de la institución, Dirección a la que pertenece.

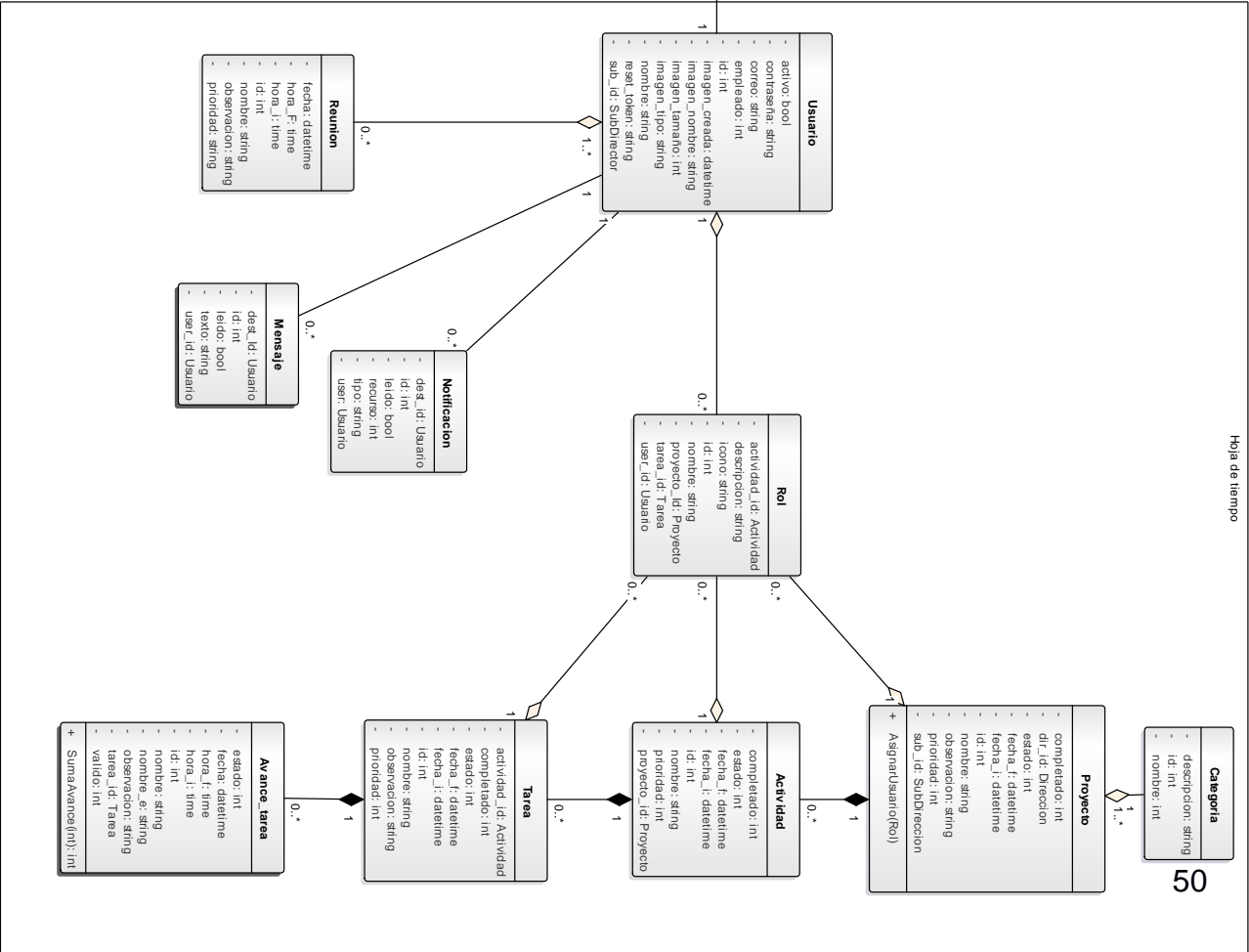
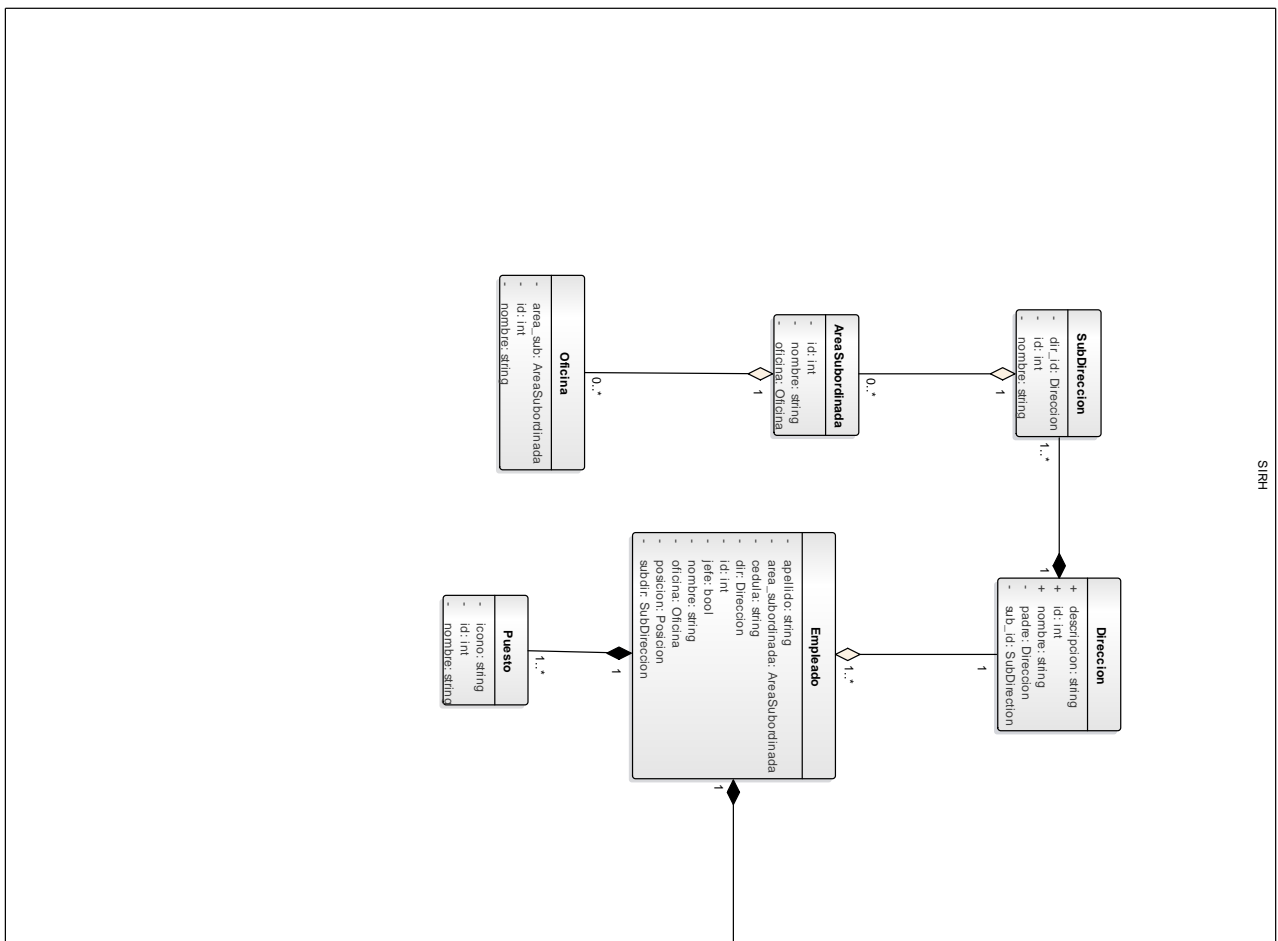


Figura 13 Diagrama de clases

6.2.3. Diseño de la base de datos

A partir del diagrama de clases se puede definir la estructura de la base de datos que se necesita para almacenar ordenadamente la información del personal de la institución *ver anexo 3*

6.2.4. Diagramas de secuencia

Los diagramas de secuencia definen las posibles vías de ejecución del software mostrando los mensajes.

6.2.4.1. Diagrama de secuencia de inicio de sesión

- Aquí la aplicación provee un token de autenticación si los datos proporcionados por el usuario son correctos.

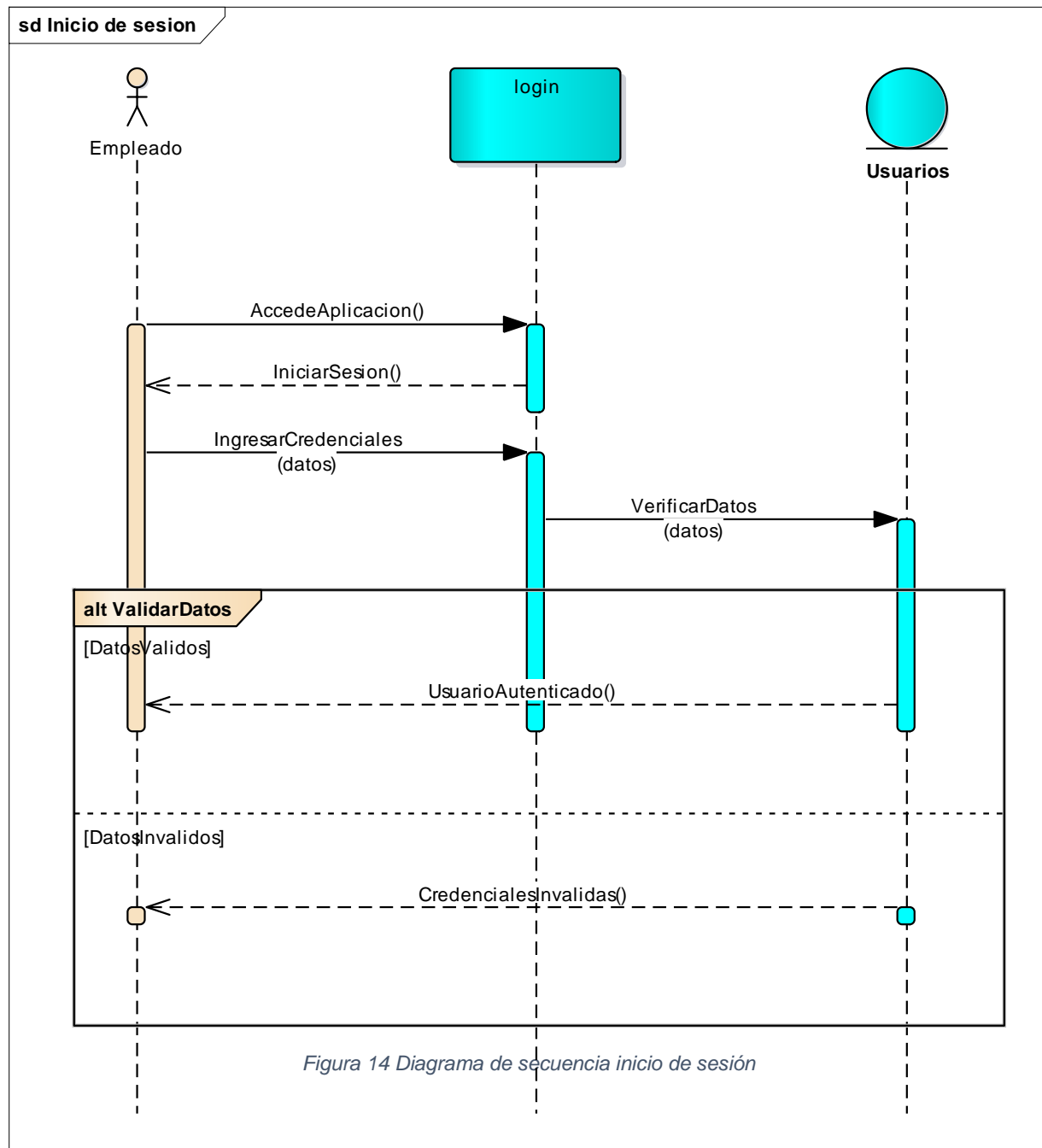


Figura 14 Diagrama de secuencia inicio de sesión

6.2.4.2. Diagrama de secuencia de recuperar contraseña:

- Se basa en enviar un token al correo del usuario para reestablecer la contraseña de la cuenta en caso de emergencia.

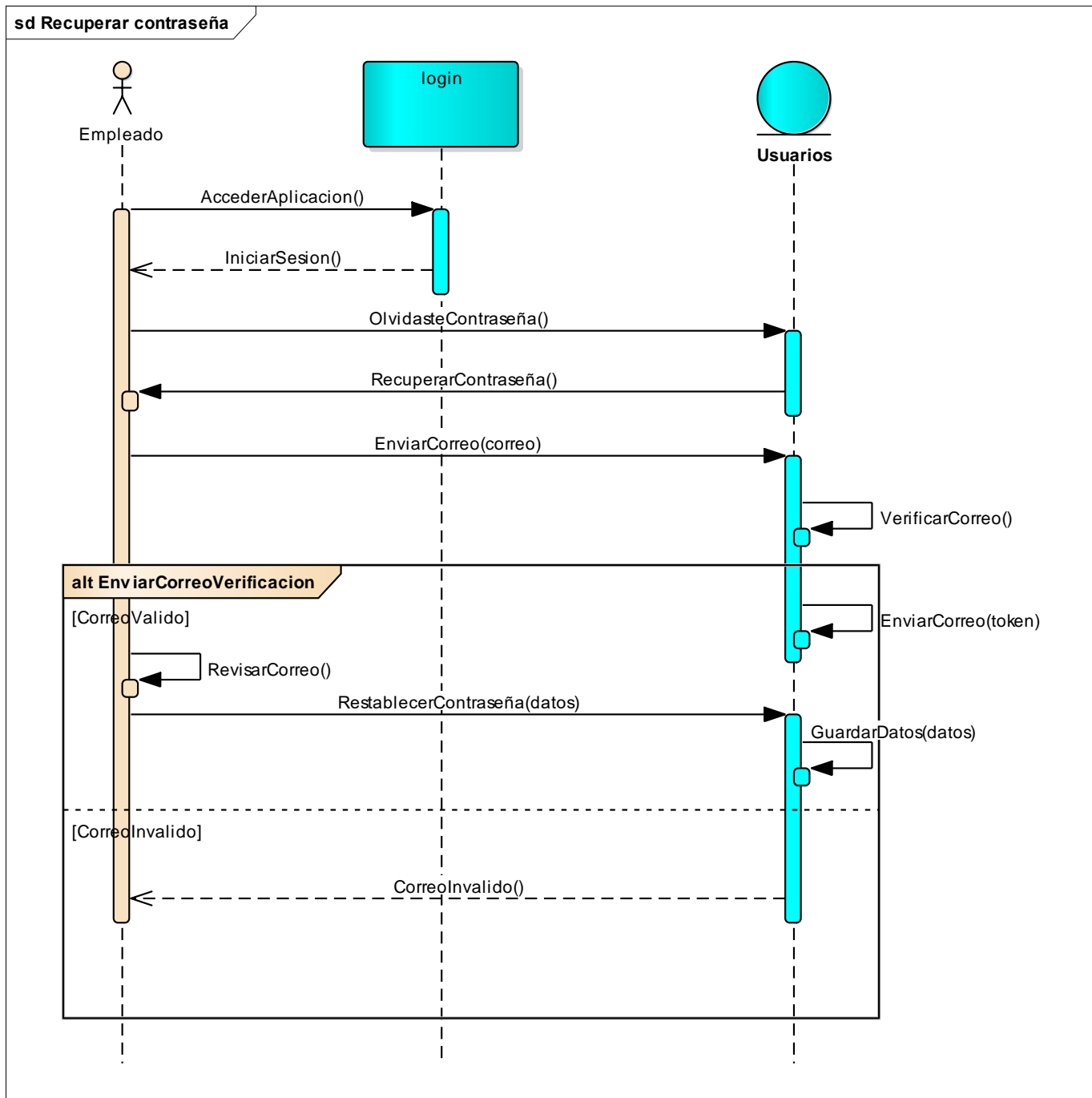


Figura 15 Diagrama de secuencia recuperar contraseña

6.2.4.3. Diagrama de secuencia de registrar empleado:

- Este método se usa para registrar un empleado al sistema una vez este se ha ingresado en el sistema de recursos humanos.

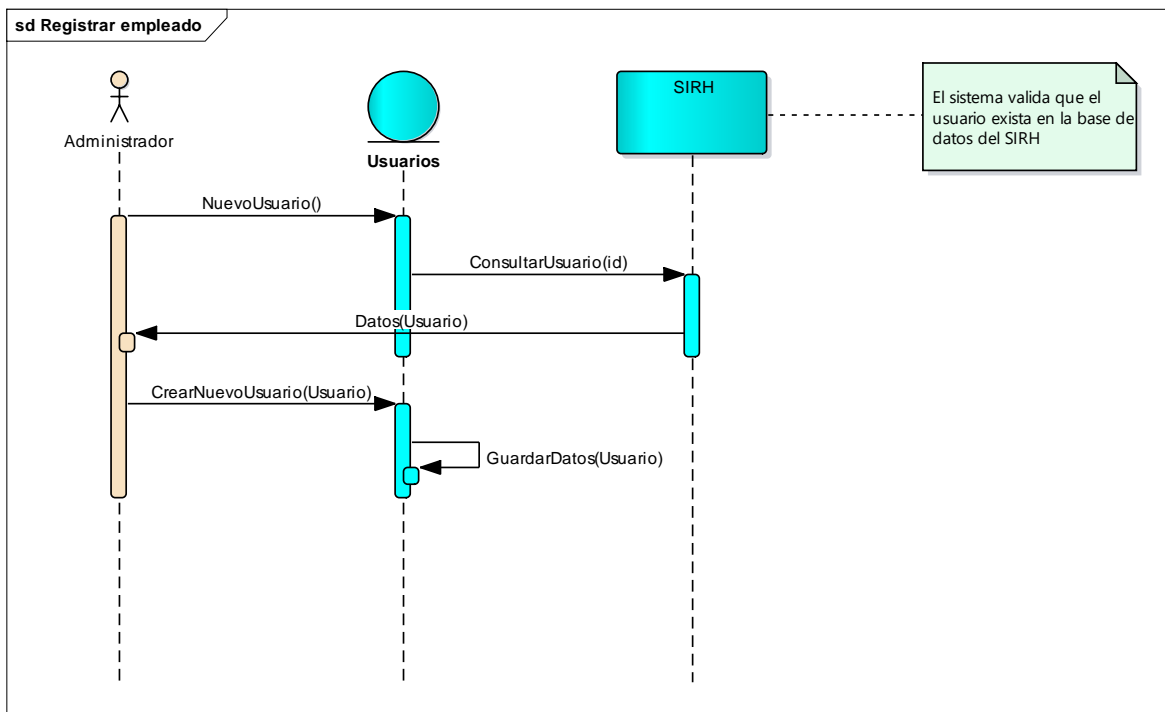
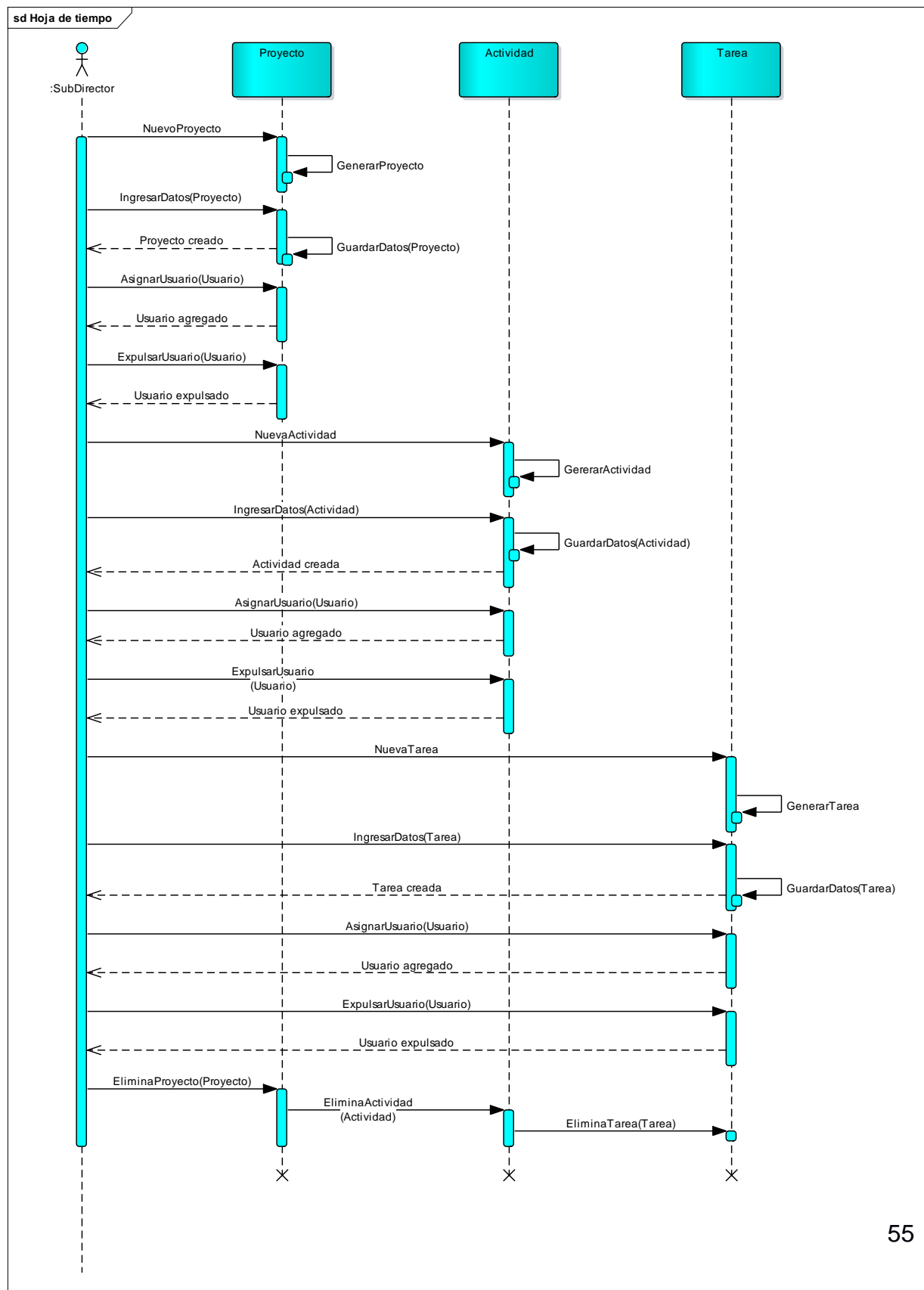


Figura 16 Diagrama de secuencia registrar empleado

6.2.4.4. Diagrama de secuencia general hoja de tiempo:

- Se crea una estructura de trabajo para comenzar a llenar el historial de las tareas de los usuarios.



6.2.4.5. Diagrama de secuencia registrar historial de tarea:

- Luego de creada la estructura de trabajo organizada los usuarios podrán hacer referencia a una tarea para ir llenando el avance que han registrado, para ser validado por sus superiores. El avance es validado de 0 a 5 estrellas según su nivel de complejidad.

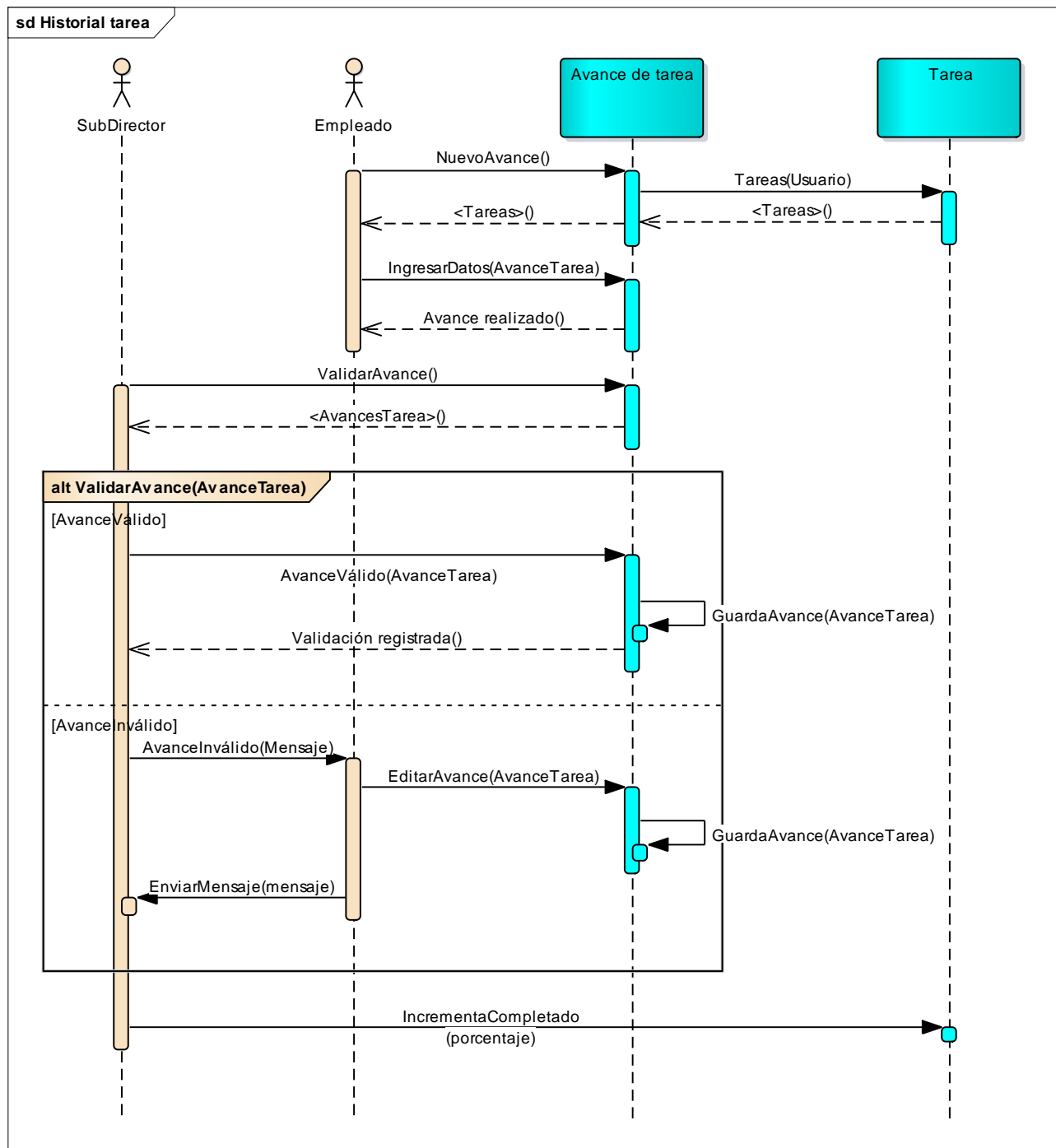


Figura 18 Diagrama de secuencia historial de tarea

6.2.4.6. Diagrama de secuencia gestionar reuniones:

- Para las reuniones se reserva un local los cuales se pueden llevar control como catálogo.

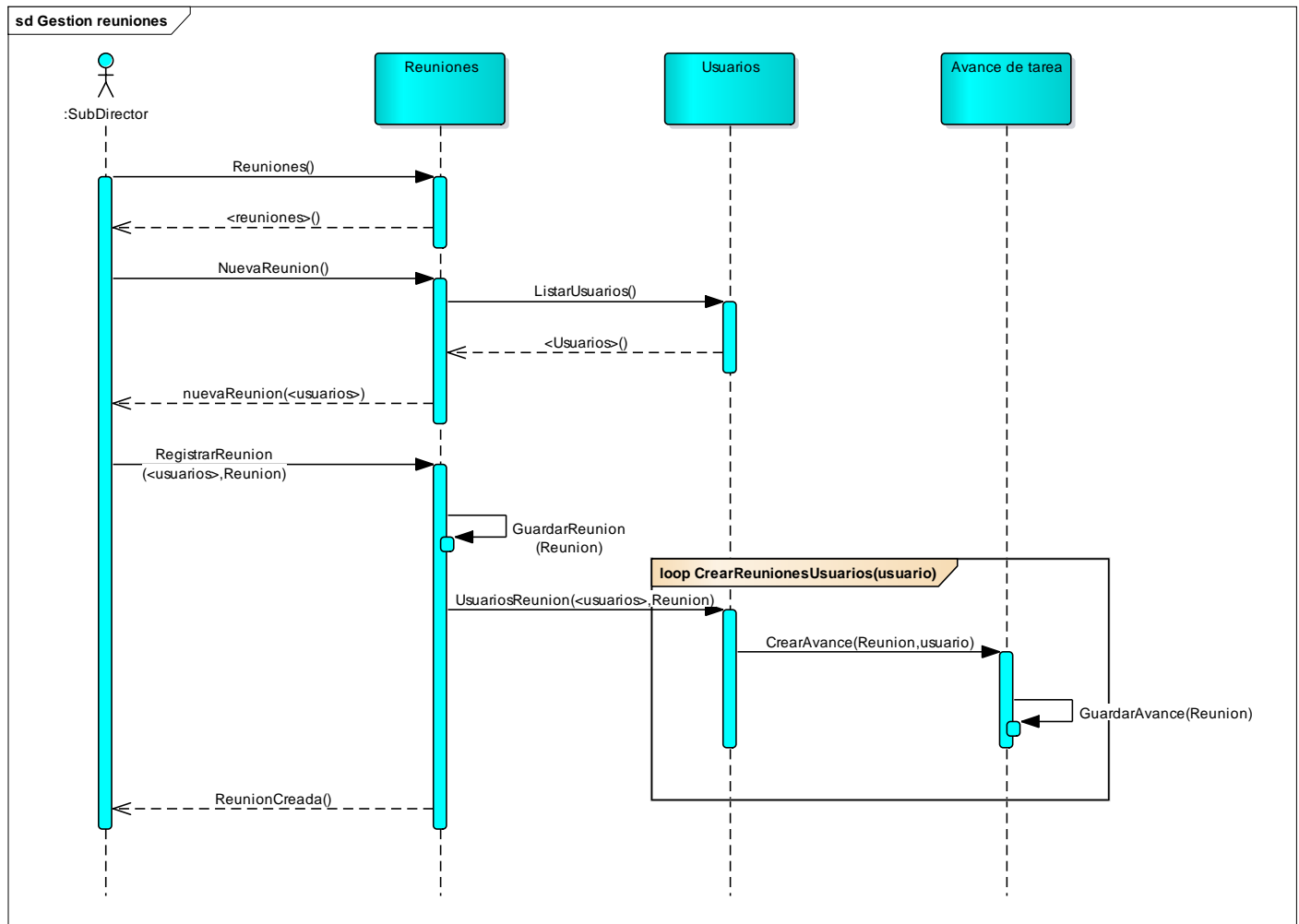
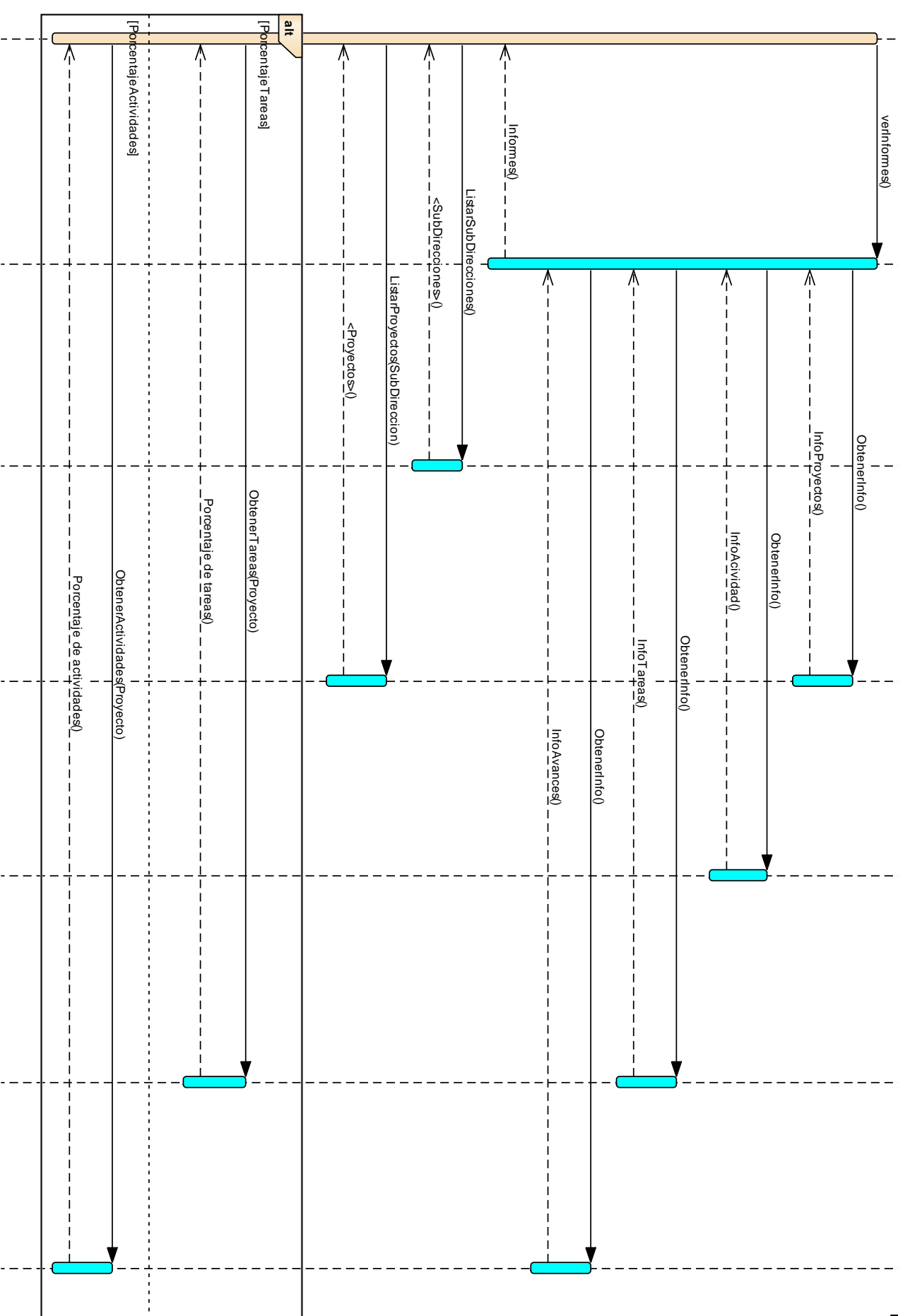


Figura 19 Diagrama de secuencia gestionar reuniones

6.2.4.7. Diagrama de secuencia ver informes:

- Este diagrama representa de manera general el panel principal para usuarios directores o subdirectores, los cuales filtran la información según tu nivel dentro del sistema. Este dashboard muestra gráficos de distintos tipos con información pertinente para ayudar y facilitar la toma de decisiones para los directivos de las áreas, o reportes dinámicos que se pueden filtrar según sea la necesidad.



6.2.5. Diagrama de despliegue

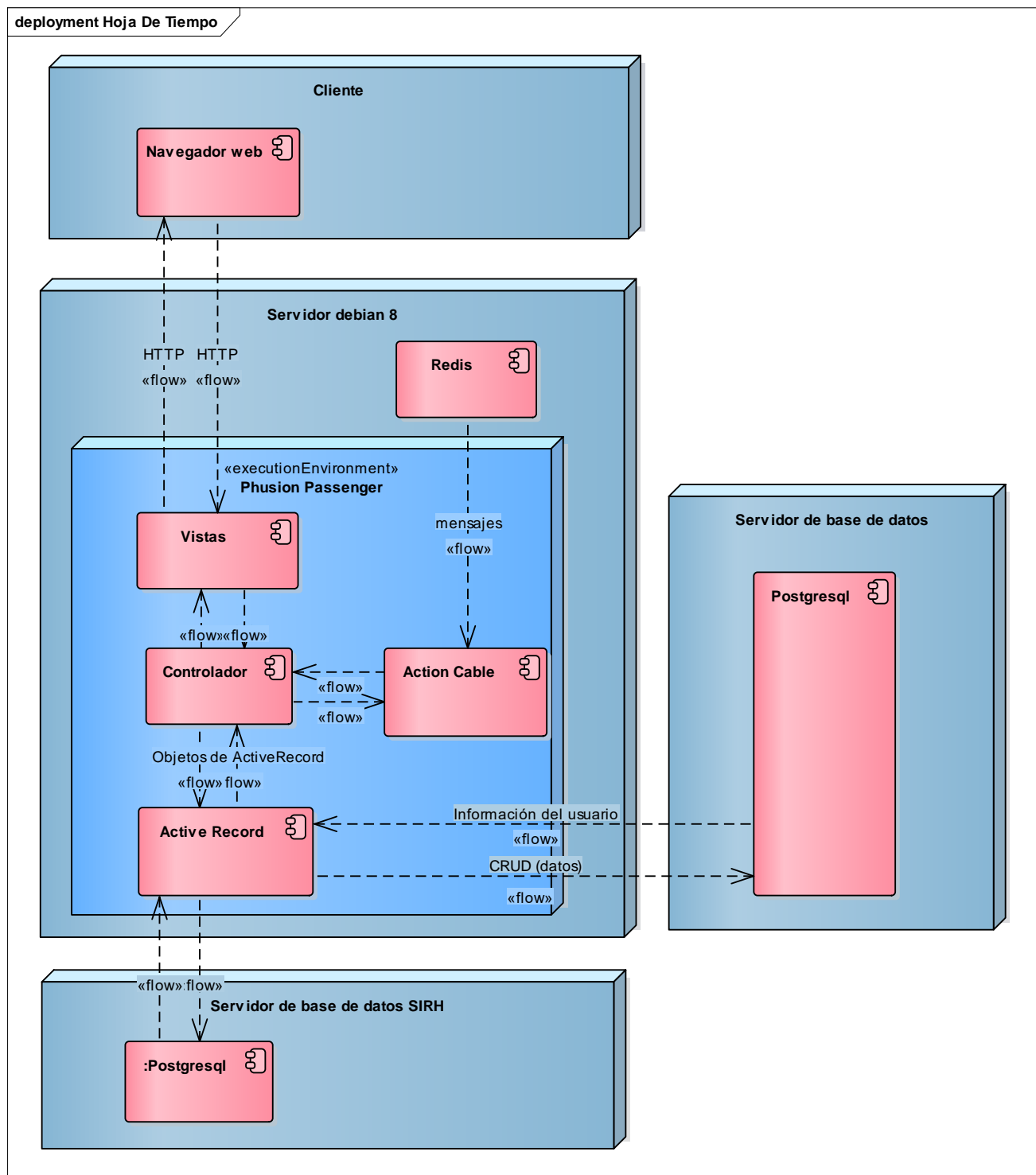


Figura 21 Diagrama de despliegue

En el diagrama de despliegue se muestran los componentes básicos de la infraestructura que se ha creado para dar solución al planteamiento de esta

monografía que es un sistema de información diseñado para ayudar en la toma de decisiones operativas para el instituto, se han abordado las configuraciones necesarias y software disponible para lograr alojar la aplicación de rails en el servidor debían y las bases de datos ya definidas que dan respuesta con la información que el cliente solicite a través del navegador web.

6.3. Desarrollo del software

El sistema tiene por objetivo llevar un mejor control del tiempo de ejecución de proyectos y del personal, ayudar en la evaluación del desempeño tanto técnico como profesional, autoevaluación de las actividades y desempeños, que pueda ser usado por otras entidades gubernamentales.

Desarrollado bajo estructura de Modelo Vista Controlador en Ruby on Rails, en el entorno del sistema operativo de Ubuntu 16.04, teniendo con PostgreSQL 9.6 y Redis 4.0 para gestionar bases de datos de registros y memoria caché utilizado para el chat en tiempo real, utilizando Atom como editor de texto, y la terminal de Ubuntu para el desarrollo de la aplicación.

Una de las ventajas de rails es que este framework gestiona una aplicación web que bajo una estructura ya conocida MVC en las cuales separan responsabilidades dentro de la aplicación. Con MVC podemos ordenar el código obteniendo un código fácilmente adaptable y gracias a las librerías o gemas que una gran cantidad de desarrolladores han contribuido se puede desarrollar código rápidamente enfocándonos más en el dominio del problema directamente.

Para la capa de modelos se cuenta con el ORM de rails llamado active record, es una herramienta muy poderosa y nos facilita el acceso a los datos dentro de la aplicación la cual podemos fácilmente manipular mediante clases y definir sus comportamientos mediante funciones, active record nos brinda mecanismos para:

- Representar modelos y sus datos.
- Representar asociaciones entre estos datos.

- Representar jerarquía de herencia a través de modelos relacionados.
- Validación de modelos antes de ser guardados.
- Mantener las operaciones orientadas a objetos mediante clases.

Los controladores son los encargados de gestionar los procesos de los usuarios, las peticiones que vienen de la vista van directamente al controlador respectivo y al método respectivo, es este que se encarga de gestionar las peticiones, realizar las operaciones deseadas y responder a la vista con la información que sea necesaria.

Para el diseño de la aplicación se ha utilizado la librería de diseño MaterializeCss que cuenta con un estilo minimalista, controles muy intuitivos y completos, diseño responsivo, es open source y es muy fácil de adaptar a cualquier necesidad del desarrollador.

El sistema también incorpora una herramienta de chat en vivo que aporta la comunicación en tiempo real implementada mediante WebSockets para ser independiente, ya que en algunas áreas del instituto por motivos de seguridad a los datos que se manejan no se cuenta con acceso a internet, WebSockets nos permite tener varios canales de comunicación para automatizar en tiempo real diferentes herramientas.

La aplicación se basa en poder crear estructuras de proyectos, actividades y tareas a las cuales van a estar disponibles para los usuarios realizar avances de tareas con respecto a una tarea, estos avances se van a ser validados por los directivos de cada usuario.

Cada usuario podrá ver su historial de avances y los usuarios de más alto nivel podrán validar los de los usuarios de más bajo nivel que estén dentro de su Dirección o Sub Dirección.

La figura 20 muestra mejor la estructura de código de la aplicación donde separa la estructura según su responsabilidad haciéndolo más ordenado y simple para el desarrollo, teniendo las carpetas necesarias para la ejecución de la aplicación y el desarrollo mediante servicios RESTFull de los cuales se encargan los controladores como podemos ver en la figura 20.

```

Project
├── hojadedetempo
│   ├── .git
│   ├── app
│   │   ├── assets
│   │   ├── channels
│   │   ├── controllers
│   │   ├── helpers
│   │   ├── jobs
│   │   ├── mailers
│   │   ├── models
│   │   ├── views
│   │   ├── bin
│   │   ├── config
│   │   ├── db
│   │   ├── lib
│   │   ├── log
│   │   ├── public
│   │   ├── test
│   │   ├── tmp
│   │   └── vendor
│   │       ├── .rake_tasks-
│   │       ├── config.ru
│   │       ├── Gemfile
│   │       ├── Gemfile.lock
│   │       ├── Passengerfile.json
│   │       ├── Rakefile
│   │       └── README.md
│   ├── reunion.html.erb
│   ├── Comentarios.js.erb
│   ├── setting_controller.rb
│   ├── reply.rb
│   ├── EditComent.js.erb
│   └── routes.rb
└── app/controllers/setting_controller.rb
    1 class SettingController < ApplicationController
    2   load_and_authorize_resource :class => TRole,
    3   :except => [:nuevousuario,:guardausuario,:updateSubDirection,
    4   :editSubDirection,:deleteSubDirection,:setActive,:setUnActive,
    5   :newsudir,:checkProgram,:getUsers,:destroydir,:destroyreunion,
    6   :editreunion,:reunion,:reunions,:updatereunion,:editdirection,:updatedirection,
    7   :direction,:nuevacategoria,:edit_category,:update_category,
    8   :deleteCategory,:createComment,:category,:index,:catch_404,
    9   :newReply,:Reply,:EditComent,:read,:usuario]
   10   skip_before_action :authenticate_user!,
   11   :only => [:nuevousuario,:guardausuario,:users]
   12   layout 'app'
   13
   14   #like
   15   def Like
   16     @like = Like.create(like_params)
   17     respond_to do |format|
   18       if @like.save
   19         format.js {render :layout => false }
   20       else
   21         format.js {render :layout => false }
   22       end
   23     end
   24   end
   25   #read path
   26   def read=
   27
   28   #post
   29   def Reply
   30     @rep = Reply.create(reply_param)
   31     @rep.user_id = current_user.id
   32     respond_to do |format|
   33       if @rep.save
   34         format.js {render :layout => false }
   35       else
   36         format.js {render :layout => false }
   37       end
   38     end
   39   end
  
```

Figura 22 Estructura de código de la aplicación

6.3.1. Análisis de riesgos y plan de contención

En este punto se ha tratado de definir algunos de los riesgos más comunes que pueden llegar a tener los procesos de la organización, con el fin de obtener un plan de contingencias para cualquier eventualidad se ha realizado la siguiente tabla de posibles riesgos que puede presentar la institución en cuanto al sistema desarrollado, a continuación, se muestran algunos riesgos, así como plan de contingencia y mitigación que se proponen para el correcto funcionamiento del sistema.²³

²³ Información base de datos fue obtenida de entrevista grabada anexada en el disco.

#	Causa	Evento	Descripción del Riesgo	Estrategias y acciones Preventivas	Contingencias	Responsable
1	Interrupción de la fuente de alimentación	El suministro de energía se puede ver afectado por la caída de un poste de luz	Un automóvil podría ocasionar este evento	El centro de datos cuenta con una fuente de alimentación independiente del instituto el cual es exclusivo y	El centro de datos cuenta con una planta eléctrica de emergencia que entra en acción automáticamente para que los servidores continúen funcionando normalmente en caso de que la fuente de alimentación falle.	-Bernardo Orozco. -Wesley Sang
2	Terremoto	Este desastre natural puede ocurrir ya que estamos en un país altamente sísmico	Movimiento telúrico de la tierra podría derribar edificios	El centro de datos está adaptado para resistir terremotos ya que está ubicado alejado de edificios altos, es un edificio aislado y con construcción antisísmica para resistir		
3	Huracán	Desastre natural muy probable debido a la ubicación y clima tropical		El edificio del centro de datos está hecho para soportar vientos huracanados		
4	Pérdida total o parcial de información	Fallo en los servidores donde está ubicado el gestor de base de datos.		Actualmente el instituto cuenta con un mecanismo de resguardo de información para las bases de datos que estén alojadas en el servidor de producción, un servidor virtual está haciendo copias cada cierto tiempo para resguardo de la información, en caso de fallo se puede perder los datos de los últimos 15 minutos.	Se cuenta con un mecanismo de respaldo de información el cual realiza un respaldo automático semanal mediante tareas programadas las cuales ejecutan un script que respalda las bases de datos el cual es pasado a discos lentos, y posteriormente llevado banco central de datos donde son almacenados, todo bajo un proceso muy riguroso ya que los accesos al datacenter del instituto son mediante huella dactilar y hay cámaras de seguridad en todo el perímetro.	-Meyling Sierra -Fabiola Castillo

Tabla 2 riesgos y contingencias

6.4. Pruebas

Se definió un plan de pruebas a ejecutar en cada iteración de la espiral de la aplicación con el objetivo de garantizar la calidad del software, la funcionalidad y la facilidad de uso del sistema, para estos casos de prueba se definió al menos 1 día de pruebas para cada módulo para asegurarnos que podemos seguir a la siguiente fase de la espiral.

Las pruebas se realizaron desde las primeras versiones de la aplicación ya sea pruebas funcionales y pruebas de ejecución del programa. Las pruebas se hacen para identificar posibles fallos de funcionamiento, configuración y usabilidad del programa.

A medida que los sistemas crecen en complejidad e interoperabilidad estos están más propensos a sufrir de fallos en el sistema o bugs, lo cual puede llevar a una situación engorrosa y fallos en el sistema.

Es por esto por lo que las aplicaciones en Ruby on Rails están enfocadas a pruebas desde su inicio lo cual podemos notar por la creación automática del folder test que se crea al iniciar la aplicación, en este folder se destina guardar archivos de prueba para controladores y modelos, escritos en ruby efectivamente. Para este proyecto usamos también de la gema “rspec” de ruby la cual es una librería muy creativa y poderosa para probar aplicaciones de ruby, las pruebas que a continuación se presentan son pruebas unitarias de código, las cuales nos garantizan la funcionalidad y calidad de la aplicación para futuras extensiones.

Como podemos observar en la figura 24, extraído del código fuente de la aplicación, este es un archivo de prueba para el controlador de proyectos en el cual se prueban todas las funciones unitariamente, o sea que cada método del controlador esté funcionando a cómo debería y este no presenta “bugs” o errores en el programa. Cada bloque de “describe” nos permite probar un método del controlador y con “it” le indicamos una acción a realizar en diferentes contextos para los métodos,

Con la gema se pueden crear modelos de pruebas denominados “factories” los cuales nos permiten rellenar modelos con información para pasarle a nuestras pruebas. Ver figura 22

```
1  FactoryBot.define do
2    factory :user do
3      email 'denisg-g@hotmail.com'
4      name 'Denis '
5      password '12345687'
6      is_active true
7    end
8  end
9
```

Figura 23 factory para login usuario

Además de probar los controladores la gema también nos permite crear pruebas para probar las validaciones en nuestros modelos como se muestra en la figura 23, aquí probamos los casos en los que las pruebas pasen y casos donde se necesitan validaciones para verificar que nuestros modelos están validándose de la manera correcta pasándole como variable a incluir mensajes de error si es el caso o verificando el estado de la petición.

```
1  require 'rails_helper'
2
3  RSpec.describe Project, type: :model do
4    describe "GET #index" do
5      it "is valid with name begdate and enddate" do
6        proy = Project.new(
7          name: 'Aaron',
8          beg_date: '2018-02-04T04:06:05',
9          end_date: '2018-09-14T04:06:05')
10       expect(proy).to be_valid
11     end
12
13     it "is invalid without begdate" do
14       proy = Project.new(beg_date: nil)
15       proy.valid?
16       expect(proy.errors[:beg_date]).to include("no puede estar en blanco")
17     end
18
19     it "is invalid without a lastname" do
20       contact = Project.new(end_date: nil)
21       contact.valid?
22       expect(contact.errors[:end_date]).to include("no puede estar en blanco")
23     end
24
25     it "is invalid without an email address" do
26       contact = Project.new(name: nil)
27       contact.valid?
28       expect(contact.errors[:name]).to include("no puede estar en blanco")
29     end
30   end
31 end
```

Figura 24 pruebas para modelo proyecto

Una vez configurado nuestro entorno de pruebas y los diferentes factories que vamos a estar usando podremos verificar que nuestros métodos estén funcionando para explicar un poco el código utilizaremos el método de crear proyecto ver figura 24. La línea “describe ‘POST create’ do” nos

```
49
50 describe 'POST create' do
51   before do
52     @attributes = {
53       name: 'tarea por realizar',
54       beg_date: '13/08/2018T04:05:21',
55       end_date: '13/08/2018T04:05:21',
56       displayed: true,
57       sub_id: 42,
58       program_id: 42,
59       technology_id: 1,
60       created_at: '13/08/2018T04:05:21',
61       updated_at: '13/08/2018T04:05:21',
62     }
63   end
64   context 'with valid params' do
65     it 'creates a new project' do
66       expect { Project.create(@attributes) }.to change{ Project.count }.by(1)
67     end
68     it 'assigns a newly created project as @project' do
69       post :create, project: @attributes
70       expect(assigns(:project)).to be_a(Project)
71     end
72     it 'redirects to the created project' do
73       post :create, project: @attributes
74       expect(response).to redirect_to(project_path(Project.last))
75     end
76   end
77 end
```

Figura 25 prueba para crear proyecto

indica que es lo que se va a probar en este caso un nuevo proyecto y se inicializa una variables con los atributos de un proyecto para que sea válido, luego se crea un contexto llamado “parámetros válidos” y en el bloque “it” se define que al crear un nuevo proyecto el número de proyectos en existencia debe aumentar en uno; otro bloque “it” nos indica que este proyecto creado debe ser almacenado en una variable de tipo “Proyecto”, y por último tenemos un bloque “it” que define nos tiene que redirigir al nuevo proyecto creado.

Todo esto con la sintaxis elegante de ruby podemos proceder a correr las pruebas para la app, si todo está bien nos muestra un mensaje en verde en el cual indica que las pruebas pasaron con éxito, de lo contrario nos hace saber que la prueba está fallando y podremos revisar y mitigar los errores uno a uno. Ver figura 24 ²⁴

²⁴ Para las pruebas hemos usado estos enlaces (Eftimov, 2016) y (Relish, s.f.)

```

Deprecated style:
get :show, { id: 1 }, nil, { notice: "This is a flash message" }

New keyword style:
get :show, params: { id: 1 }, flash: { notice: "This is a flash message" },
  session: nil # Can safely be omitted.
(called from block (4 levels) in <top (required)> at /home/denis/Documentos/hojadetiempo/spec/controllers/work_controller_spec.rb:90)
DEPRECATION WARNING: Using positional arguments in functional tests has been deprecated,
in favor of keyword arguments, and will be removed in Rails 5.1.

Deprecated style:
get :show, { id: 1 }, nil, { notice: "This is a flash message" }

New keyword style:
get :show, params: { id: 1 }, flash: { notice: "This is a flash message" },
  session: nil # Can safely be omitted.
(called from block (4 levels) in <top (required)> at /home/denis/Documentos/hojadetiempo/spec/controllers/work_controller_spec.rb:90)
DEPRECATION WARNING: Using positional arguments in functional tests has been deprecated,
in favor of keyword arguments, and will be removed in Rails 5.1.

Deprecated style:
get :show, { id: 1 }, nil, { notice: "This is a flash message" }

New keyword style:
get :show, params: { id: 1 }, flash: { notice: "This is a flash message" },
  session: nil # Can safely be omitted.
(called from block (4 levels) in <top (required)> at /home/denis/Documentos/hojadetiempo/spec/controllers/work_controller_spec.rb:95)
DEPRECATION WARNING: Using positional arguments in functional tests has been deprecated,
in favor of keyword arguments, and will be removed in Rails 5.1.

Finished in 1.05 seconds (files took 1.95 seconds to load)
32 examples, 0 failures

denis@denis-ThinkPad-T430:~/Documentos/hojadetiempo$

```

Figura 26 salida de pruebas

6.5. Implementación

El en servidor se instaló Ruby 2.4.2, Rails 5.0.6, corriendo en un servidor Passenger 5.1.12 y redis-cli 4.0.2 para la memoria caché en el puerto 6379, conectándose al servidor de base de datos para acceder a la información como lo muestra el diagrama de despliegue *ver figura 22*, estos tienen que estar 100% disponibles para evitar que los datos sean comprometidos, el servidor recibe las peticiones de los clientes por el puerto 443; el servidor usa un certificado comprado por la institución

Para el despliegue del sistema se debe acceder al servidor mediante Secure Shell (SSH) el cual nos permite acceder al sistema mediante la terminal con claves públicas y privadas las cuales se han generado para asegurar la privacidad de la conexión y garantizar la seguridad de la aplicación. A continuación, se definen una lista de pasos para el despliegue de la aplicación en un servidor Debian 8 estos

pasos se pueden usar para desplegar la aplicación en otro servidor o ya sea otra aplicación de Ruby on Rails, esta guía se ha hecho basado en las siguientes páginas: (Ruby Org, 2017), (Passenger, s.f.).

Paso 1: Primeramente, se tiene que actualizar el sistema operativo y descargar las dependencias necesarias para la instalación de las herramientas necesarias para el desarrollo con el siguiente comando:

```
$ sudo apt-get update && sudo apt-get upgrade  
$ sudo apt-get install zlib1g-dev libssl-dev libreadline-dev  
libgdbm-dev openssl
```

Paso 2: Creamos una carpeta donde vamos a instalar Ruby y todos sus componentes en este caso lo hacemos en la carpeta Home, descargamos los binarios de Ruby desde el repositorio oficial y procedemos a desempaquetar e instalar en nuestro sistema.

```
$ mkdir Ruby24 && cd Ruby24  
/Ruby24 $ wget https://cache.ruby-  
lang.org/pub/ruby/2.4/ruby-2.4.0-rc1.tar.gz  
/Ruby24 $ tar -xzf ruby-2.4.0-rc1.tar.gz  
/Ruby24 $ ./configure  
/Ruby24 $ make  
/Ruby24 $ sudo make install
```

Paso 3: Ahora procedemos a instalar gemas o librerías necesarias para nuestro entorno de desarrollo de open source y para poder usar el framework Rails.

Con el primer comando instalamos la gema de bundler una herramienta para administrar las dependencias y las gemas, el segundo comando actualizamos el sistema e instalamos algunas librerías de desarrollo y NodeJs que va a servirnos con Rails para trabajar nuestros scripts del lado del cliente.


```
$ gem install bundler --no-rdoc --no-ri
$ sudo apt-get update && sudo apt-get install -y apt-transport-https
ca-certificates && curl --fail -ssl -o setup-nodejs
https://deb.nodesource.com/setup_8.x && sudo bash setup-nodejs && sudo
apt-get install -y nodejs build-essential
```

Paso 4: Instalar el servidor Passenger, primero accedemos a la aplicación que primeramente copiamos a través de FTP o SCP en mi caso mediante SCP con un explorador de archivos de Ubuntu y lo he alojado en la dirección [/var/www/hojadetiempo/](#), buscamos el archivo Gemfile y nos aseguramos de eliminar cualquier otro servidor (unicorn, thin o puma) y nos aseguramos de escribir la siguiente línea:

```
gem "passenger"
```

Paso 5: Cuando lo tengamos lo siguiente será instalarlo mediante el comando bundle, pero primero creamos un usuario de la aplicación, en nuestro caso le vamos a asignar el nombre "timesheet" que es "hoja de tiempo" en inglés, y asignamos los permisos necesarios para la aplicación:

```
$ sudo adduser timesheet
$ sudo chown timesheet: /var/www/hojadetiempo
```

Luego de esto accedemos como usuario de la aplicación y procedemos a instalar todas las gemas incluyendo Passenger:

```
$ sudo -u myappuser
$ bundle install --deployment --without development test
```

Paso 6: Lo siguiente es instalar Redis la base de datos de memoria caché open source, para esto vamos a actualizar el equipo nuevamente, luego vamos a instalar tcl un paquete necesario.

```
$ sudo apt-get update
$ sudo apt-get install tcl8.5
```

Descargamos, desempaquetamos y nos movemos a ese directorio:

```
$ curl -O http://download.redis.io/redis-stable.tar.gz
$ tar xzf redis-stable.tar.gz
$ cd redis-stable
```

Procedemos a compilar, probar e instalar Redis con los siguientes comandos:

```
/redis-stable $ make
/redis-stable $ make test
/redis-stable $ sudo make install
```

Una vez finalizado el instalador Redis viene integrado con un script para correr en modo de demonio, para acceder al script nos movemos a la carpeta utils y luego corremos el script y posteriormente probamos si salió todo bien con la instalación:

```
/redis-stable $ cd utils
/redis-stable/utils $ sudo ./install_server.sh
/redis-stable/utils $ redis-cli
redis 127.0.0.1:6379> ping
PONG
```

Mientras el script se ejecuta se pueden dejar las opciones por default solo presionando la tecla enter y una vez termina el script Redis ya estará corriendo en modo demonio.

Paso 7: Se crea una llave única secreta que encripta la sesión de usuario, que se guarda en config/secrets.yml para generar la clave se escribe el siguiente comando:

```
$ bundle exec rake secret
```

El comando genera una clave la cual copiamos y luego abrimos el archivo config/secrets.yml y nos quedara algo asi:

```
production:
  secret_key_base: 851726347630a8225c083e872d866da2cb4aac1f05c083d886
```

Paso 8: Compilar los assets o los recursos de la app:

```
$ bundle exec rake assets:precompile db:migrate RAILS_ENV=production
```

Paso 9: Inicar la app en Passenger, ahora que tenemos listo todo el entorno configurado para la aplicación, se debe crear un archivo de configuración para Passenger llamado PassengerFile.json dentro de la aplicación:

```
$ cd /var/www/hojadetiempo
$ nano PassengerFile.json
```

El cual tiene que quedar de esta manera:

```
{
  // Run the app in a production environment. The default value is
  "development".
  "environment": "production",
  // Run Passenger on port 80, the standard HTTP port.
  "port": 443,
  "ssl": true,
  "ssl_certificate": "/etc/nginx/ssl/nginx.crt",
  "ssl_certificate_key": "/etc/nginx/ssl/nginx.key",
  // Tell Passenger to daemonize into the background.
  "daemonize": true,
  // Tell Passenger to run the app as the given user. Only has effect
  // if Passenger was started with root privileges.
  "user": "timesheet",
  "unlimited_concurrency_paths": [
    "/cable"
  ]
  "nginx_config_template": "/var/www/hojadetiempo/config.erb"
}
```

Salimos con Ctrl+c y ahora con el siguiente comando iniciamos el servidor dentro de la carpeta de la aplicación:

```
/var/www/hojadetiempo $ sudo bundle exec passenger start
```

Con esto tendremos la aplicación corriendo como hemos configurado y con todos los componentes necesarios para su mejor funcionamiento

Para la implementación se creó un plan de capacitación semanal por Dirección el cual se llevará a cabo mediante una reunión de inducción donde se explicará de manera general el funcionamiento del sistema. Además, se darán manuales de usuario y tutorías especializadas para el personal de cada Dirección en la semana que le corresponda para dejar claro el funcionamiento del sistema.

7. Evaluación de resultados

Este capítulo se evalúa el desempeño de la aplicación, calculando los tiempos y esfuerzos para la creación del plan operativo anual, tomando como punto de referencia el modelo de planificación antiguo y haciendo un enfoque en las nuevas herramientas que el sistema provee en la ayuda de toma de decisiones para los directivos.

Se evalúa también el uso del software en las diferentes direcciones donde se va implementando mediante métricas establecidas.

7.1. Métricas

Se identificaron las dos métricas más importantes para evaluar las cuales son desempeño y facilidad de uso. Además de sus variables se tomaron las pruebas necesarias para comprobar los beneficios del sistema y su aporte a la institución.

- Desempeño: para evaluar el desempeño se tiene que estudiar la latencia del sistema mediante el tiempo de respuesta.
 - Tiempo de respuesta (t): tiempo requerido para generar un reporte, una función, y diferentes funcionalidades del sistema.
 - Razón de rendimiento (n): Cociente entre el tiempo de respuesta del sistema y del proceso antiguo de generación de reportes. Esta métrica

permite comparar los procesos: si es mayor a 1, entonces el sistema es mejor que el proceso antiguo. Si es igual a 1 entonces el sistema y el proceso son iguales. Y si es menor que 1 el sistema es inferior en rendimiento.

- Mejora de rendimiento porcentual ($n\%$): se define como $(n - 1) * 100\%$. Expresa en qué medida es mejor el proceso actual y el sistema.

A continuación, se definen las ecuaciones de estas variables. Nótese que, para los efectos de este proyecto, se empleará la segunda expresión en el cálculo de n .

$$n = \frac{R_{nuevo}}{R_{anterior}} = \frac{t_{anterior}}{t_{nuevo}}$$

Ecuación 1: Razón de Rendimiento

$$n_{\%} = (n - 1) * 100\%$$

Ecuación 2: Ventaja neta porcentual de rendimiento

- Facilidad de uso: Para evaluar este concepto se tomaron en cuenta las siguientes variables el esfuerzo del usuario y la simplicidad de interacción con el sistema.
 - Pasos de procedimientos: Cantidad de pasos de procedimientos necesarios para lograr una tarea determinada. Variable numérica entera.
 - Simplicidad: Grado de simplificación y adecuación al diseño que facilite la interacción con el sistema. Variable discreta con valores simple: simple, complejo.
- Estabilidad: para evaluar la estabilidad del servidor se ha ocupado una herramienta llamada webserver stress tool 8, la cual simula el acceso de un grupo determinado de usuarios al servidor para probar la red y el acceso al sistema

7.2. Método de evaluación

Para la evaluación de las métricas de desempeño se usará el método de pruebas a los procesos y partes del sistema que sean necesarios. Mientras para la facilidad de uso se usará el método de consulta.

Se definieron que las funciones más importantes dentro de la planificación operativa del instituto:

1. Creación de estructura de trabajo.
2. Reporte anual de planificación operativa.
3. Validación de un día de avance de tareas.

Se definieron del personal que van a ejecutar las pruebas para cada proceso. Para esta tarea se definieron a los directores, Sub directores y empleados del área de Sistemas Geo informáticos para estas taras de pruebas al sistema.

7.2.1. Procedimiento para evaluar la creación de una estructura de trabajo

- Se definieron pasos para pruebas de creación de una estructura de trabajo con la que actualmente se trabaja en la institución.
- Se realizaron pruebas de creación de esta estructura en el sistema y de la manera tradicional y se cronometró el tiempo y la cantidad de pasos que toma para crear los informes, y su simplicidad.
- Se tabularon los datos obtenidos de las pruebas para poder obtener los resultados y verificar si es viable el sistema.

7.3. Resultados obtenidos para la creación de estructura de trabajo

7.3.1. Evaluación de desempeño

La siguiente tabla resume los resultados de las mediciones de crear una estructura de trabajo y realizar un reporte de actividades. Tomadas para comparar la

evaluación de la razón de rendimiento (n) y mejora neta porcentual de rendimiento (n%), las cuales establecen en qué medida es mejor un sistema que el método convencional.

Tabla 3 Razón de desempeño

Caso de prueba	Método convencional	Sistema hoja de tiempo	n	n%
	t(s)	t(s)		
Prueba 1	2,100	1,600	1.3125	31.2%
Prueba 2	3,600	2,100	1.7142	71.4%
Prueba 3	2,750	2,047	1.3434	34.3%
Media	2,816.6	1,915.6	1.4567	45.6%

En la tabla podemos observar que el promedio de tiempo con el método manual o en hojas de cálculo de Excel el tiempo promedio es de 2,800 segundos y con el sistema el tiempo se reduce a 1,900 es una mejora notable en el procedimiento de creación de reportes.

7.3.2. Evaluación de facilidad de uso

Para medir la facilidad de uso lo haremos midiendo la simplicidad de uso para distintos usuarios y consultando con el usuario (simplicidad de uso) y la cantidad de esfuerzo requerido para la creación de la estructura de trabajo.

7.3.2.1. Simplicidad de uso

Para la siguiente tabla resume el grado de simplicidad de las interfaces gráficas del sistema según la precepción de cada usuario, para esto se ha entrevistado a usuarios y se evidencia que el sistema es muy simple, fácil de usar y de entender.

Tabla 4 Facilidad de uso

Caso de prueba	Método convencional	Sistema hoja de tiempo
	Simplicidad { complejo, normal, simple }	
Prueba 1	Normal	Simple
Prueba 2	Complejo	Media
Prueba 3	Normal	Normal
Media	Complejo	Simple

7.3.2.2. Pasos de procedimientos

Para este caso de prueba se han especificado que el usuario tiene que crear la cantidad de 10 actividades y 10 tareas para el proyecto a crear y la cantidad de pasos de procedimientos casi se iguala porque en Excel no hay muchos pasos más que agregar los datos a la hoja de cálculo.

Tabla 5 Pasos de procedimiento

Caso de prueba	Método convencional	Sistema hoja de tiempo
	Cantidad de pasos	
Prueba 1	24	22
Prueba 2	24	22
Prueba 3	24	22
Media	24	22

Así logramos obtener un mayor desempeño y también asegurar la facilidad de uso de la herramienta

7.4. Resultados obtenidos para el reporte de planificación

7.4.1. Evaluación de desempeño

La tabla 4 resume los resultados de las pruebas que se realizaron en lo que respecta al tiempo de descarga de un plan operativo, como no existía un sistema que hiciera esta función se va solo a representar el sistema de hoja de tiempo con fines ilustrativos.

Tabla 6 Tiempos para reporte anual

Caso de prueba	Sistema hoja de tiempo		
	Hora inicio	Hora fin	Tiempo (s)
Prueba 1	09:40:20	09:40:24	4
Prueba 2	09:41:45	09:41:47	2
Prueba 3	09:42:30	09:42:32	2.5
Prueba 4	09:44:12	09:44:15	3
Prueba 5	09:45:56	09:45:58	2.4

El resultado final nos muestra que en el proceso solo se necesitan de pocos segundos para poder obtener el informe con la información anual, el mismo se puede descargar para casos necesarios ya que se pretende ahorrar estas impresiones mientras no sea necesaria su impresión.

7.4.2. Evaluación de la facilidad de uso

7.4.2.1. *Simplicidad de uso*

Para esta prueba se obtuvieron de la misma ficha de registro la cual indica que la manera de generar reportes es la adecuada haciendo del sistema una herramienta fácil de usar y accesible para los usuarios.

En todos los resultados los usuarios pudieron realizar las tareas sin ningún problema, calificando el sistema como: “una herramienta muy buena e intuitiva para llevar el control de actividades de los usuarios”.

Todas las personas marcaron la casilla simple en esta prueba así que no vamos a tabular estos resultados, pero vamos a decir que obtuvimos un 100% de simplicidad de uso de la herramienta.

7.5. Resultados de evaluación de estabilidad

Para esta prueba hemos descargado y configurado la herramienta webserver stress tool para simular un escenario de alta disponibilidad en momentos de estrés para el servidor, se han realizado diferentes tipos de prueba que la herramienta nos brinda, como “click por minuto” y “rampa” durante un tiempo especificado con diferentes números de usuarios.

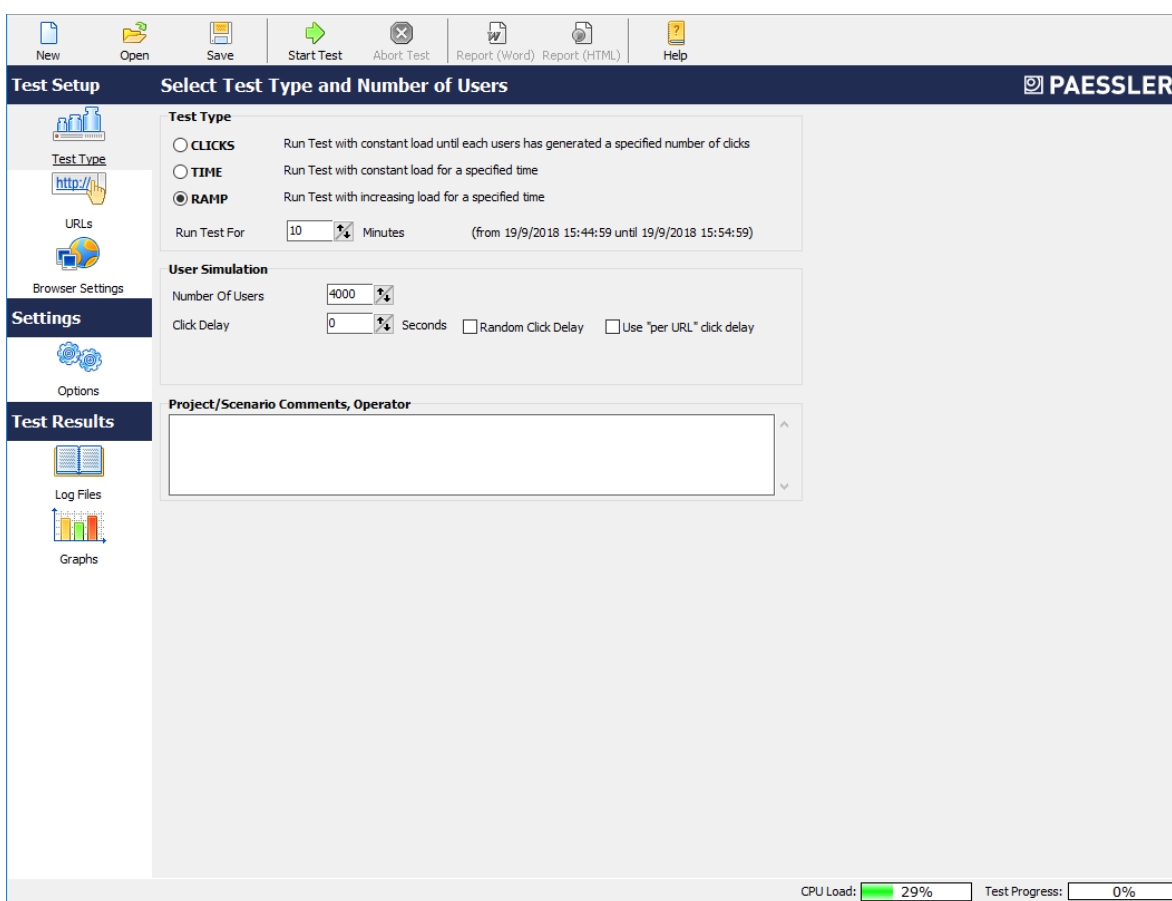


Figura 27 configuración de prueba de estrés

Especificando la opción para listar las tareas del usuario.

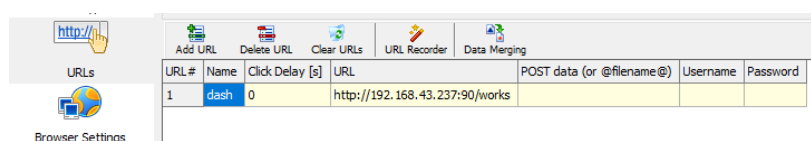


Figura 28 url configuración

Una vez configurado todo procedemos a iniciar la prueba con lo cual el programa procede a crear una cola de espera para acceder al servidor y responder a las respectivas peticiones.

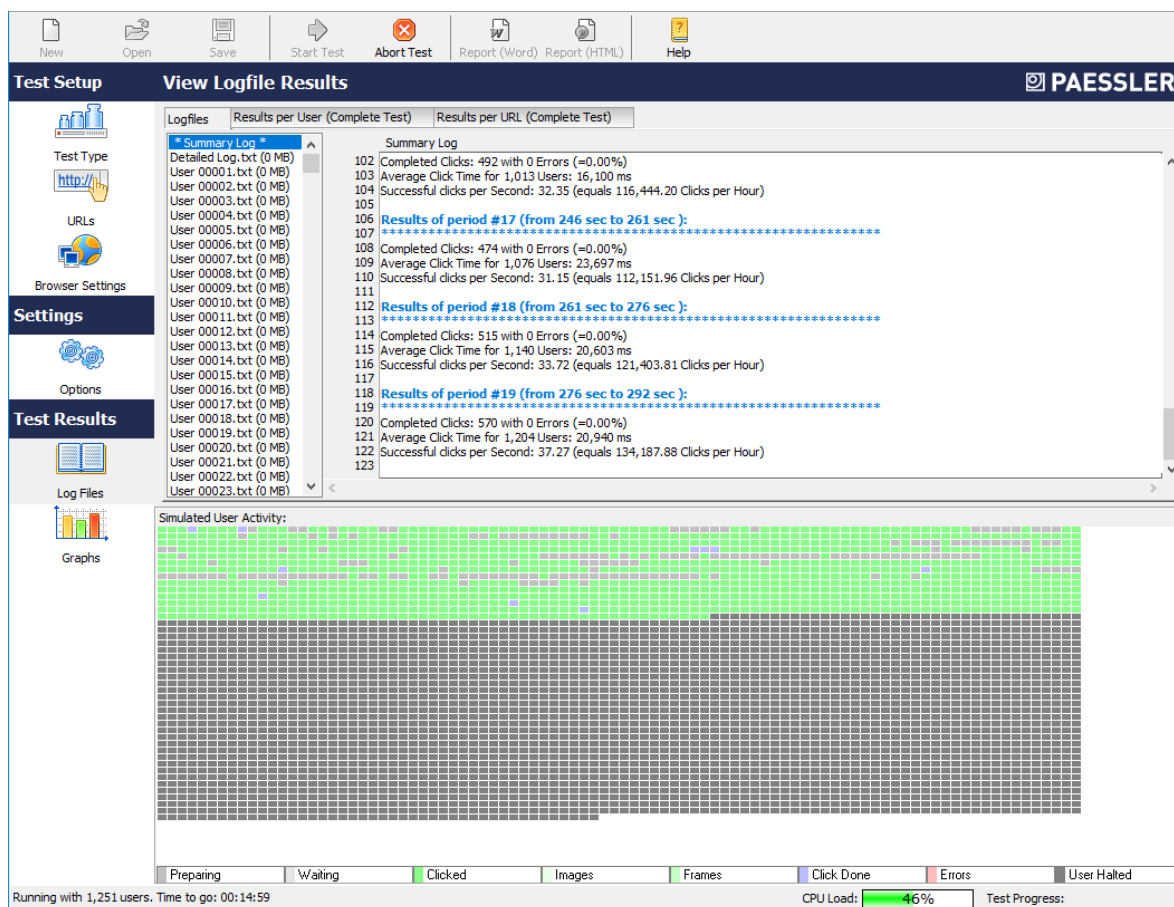


Figura 29 Ejecución de prueba de estrés

Una vez iniciado el proceso de prueba podemos verificar el uso de los recursos físicos del servidor usando la herramienta top y podemos observar que el servidor mantiene la estabilidad del sistema.

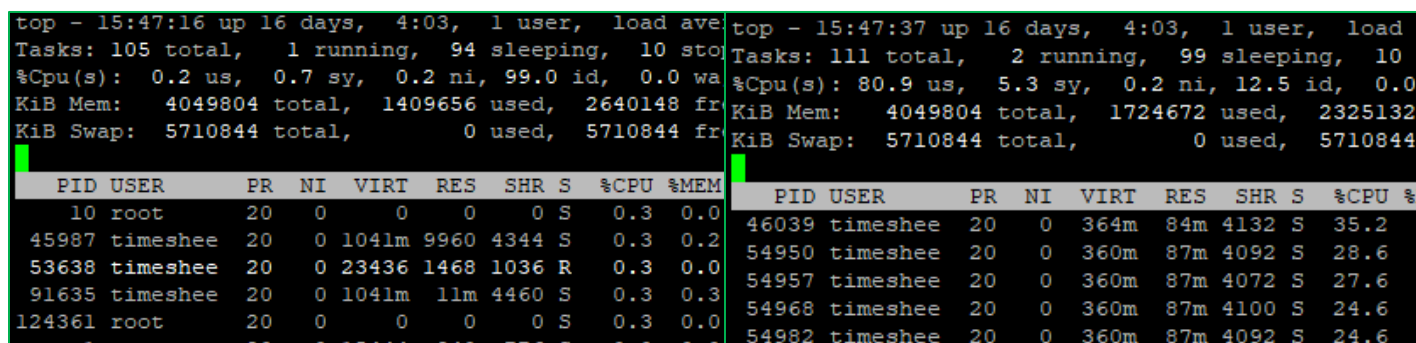
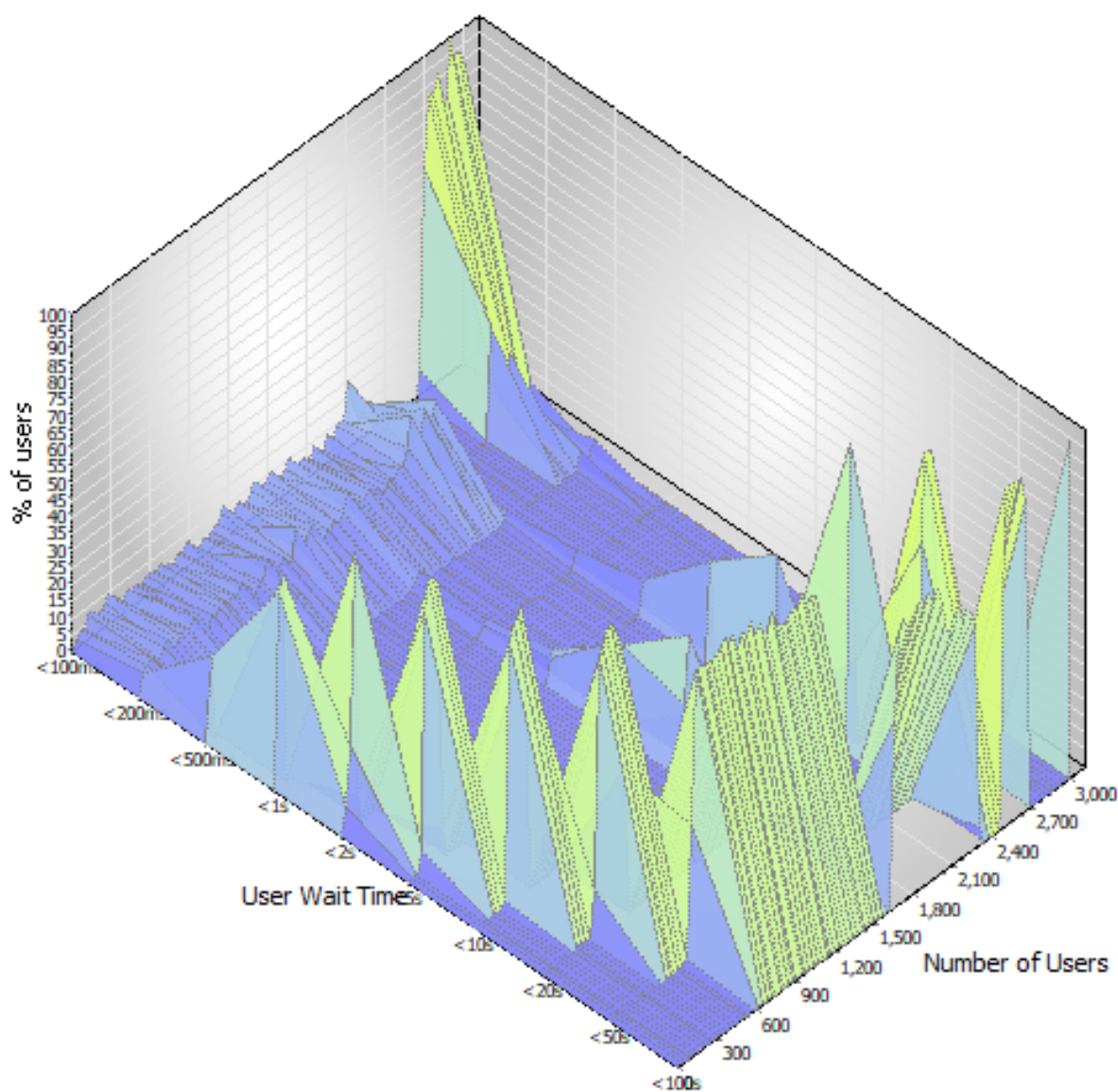


Figura 31 antes de ejecutar la prueba

Figura 31 Durante la prueba

Una vez terminada la prueba la herramienta nos brinda un reporte web con toda la información recopilada del servidor y gráficos que pueden servir para determinar el ancho de banda alcanzado, probando los límites del servidor, tiempos de respuestas, máximo de respuestas, etc.



Note: Only first 1000 items were graphed
Test Type: RAMP (run test for 10 minutes)
User Simulation: ramp test with up to 3,000 simultaneous users - 0 seconds between clicks

Figura 32 Espectro de tiempo de click

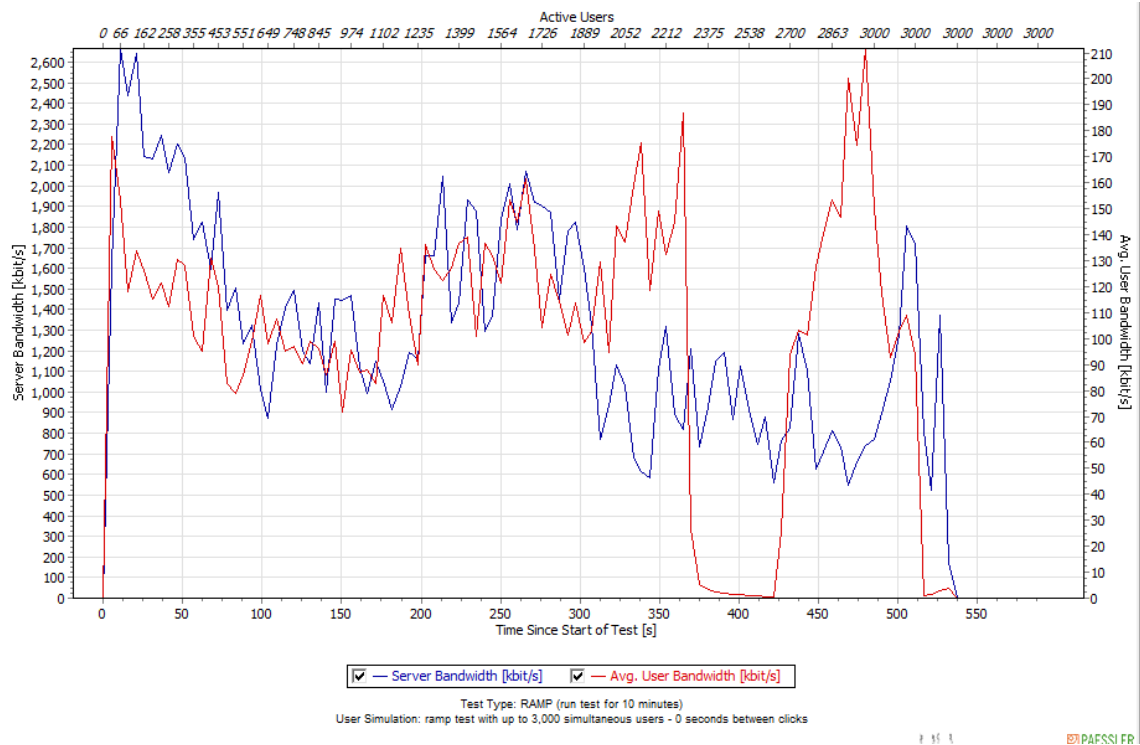


Figura 34 Ancho de banda de servidor y usuario

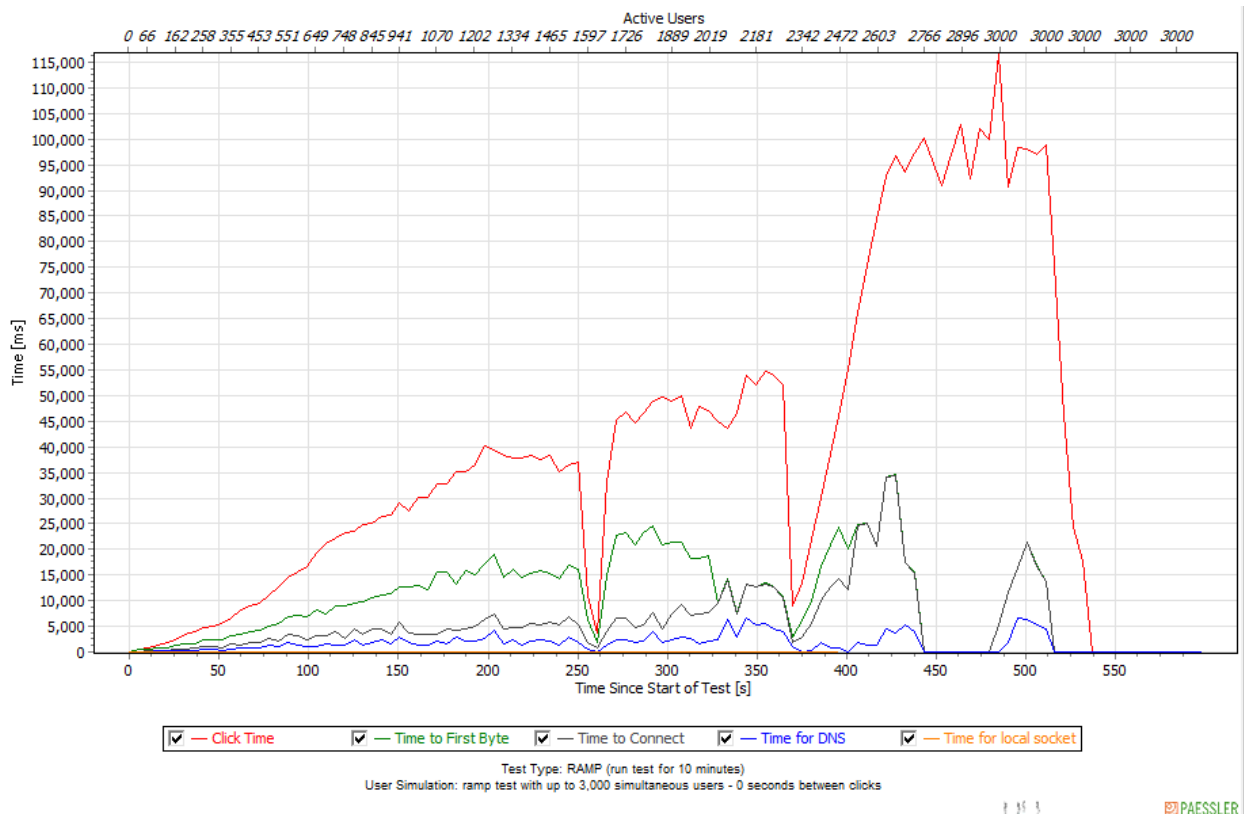


Figura 33 Tiempo de protocolo

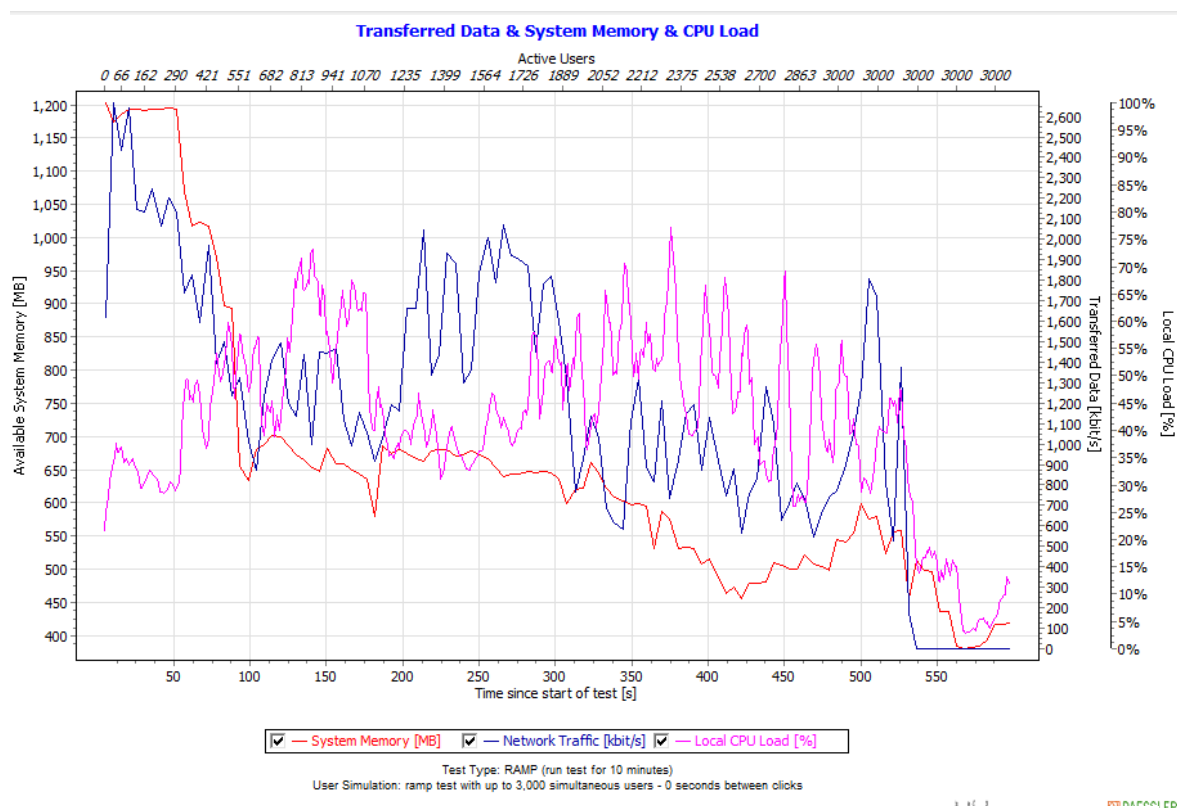


Figura 35 Recursos del servidor

Hemos llevado al servidor hasta un 85% de uso de recursos con un total de 2,000 usuarios, el servidor ha respondido satisfactoriamente a todas las peticiones, y según el plan informe de trabajo de la dirección de sistemas geo informáticos (ver anexo 8), se general aproximadamente 370 actividades para un proyecto en la dirección, tomando en cuenta que se crean en promedio 4 proyectos al año dependiendo de la dirección podemos calcular en las 8 direcciones de ineter unas 5,540 actividades en total.

Teniendo en cuenta que existen aproximadamente 515 usuarios activos en el instituto el servidor puede gestionar sus peticiones sin ningún problema.

8. Conclusiones y recomendaciones

8.1. Conclusiones

Se logró obtener toda la información relevante acerca del plan operativo anual y realizar un análisis de las necesidades y requerimientos del instituto para la creación e implementación de un nuevo sistema de auto seguimiento y planificación operativa.

Se definieron las restricciones, roles, tipos de usuarios y metodologías a utilizar de acuerdo a las normativas del instituto, para garantizar la implementación del sistema.

Se diseñó el nuevo sistema basado en la metodología de espiral y con herramientas UML se definieron los diferentes procesos que se implementaron posteriormente en el código.

Se realizó con éxito el desarrollo de la aplicación la cual consta de diferentes herramientas pensadas para aligerar y facilitar el auto seguimiento y planificación operativa del personal de la institución.

Se configuró el servidor que va a alojar la aplicación, se desplegó la aplicación y se monitoreó constantemente para verificar que todo esté en orden.

En todas las pruebas realizadas se evidencia la mejora notable de tiempos de respuesta de hasta un 45%, y con la que es más fácil simple e intuitivo el seguimiento a actividades y planificación operativa.

8.2. Recomendaciones

Una vez terminada la aplicación está abierta a sufrir nuevos cambios o incorporación de nuevos métodos y herramientas, para esto se ha creado esta sección donde se analizan posibles ajustes y se dan recomendaciones en base a lo recopilado y observado durante todo el ciclo de desarrollo para el futuro del sistema.

Una de las características más curiosas de esta implementación es que el sistema accede a 2 bases de datos según es necesario en cada petición, debido a que la información del sistema de recursos humanos (SIRH) es necesaria para los roles del sistema. Por esto se recomienda crear APIS en el sistema de recursos humanos y en hoja de tiempo para que estos puedan comunicarse y garantizar el máximo rendimiento de la aplicación.

Se recomienda también implementar servicios en la misma API de hoja de tiempo que sirvan de comunicación entre el Sistema Administrativo Financiero (SIAF) de la Dirección Administrativa Financiera (DAF) para obtener la información financiera de cada proyecto y se pueda gestionar mediante el sistema también los costos, así como otros recursos de manera relacionada de los proyectos.

Por el momento se usa WebSocket para automatizar el tiempo real en la entrega de los mensajes, pero para el futuro se pueden automatizar también las notificaciones, comentarios y “favorito” para obtener una experiencia de usuario más consistente.

Se recomienda continuar con el desarrollo de la herramienta, a fin de que pueda ser optimizada y usada por otras instituciones gubernamentales con los mismos tipos de estructuras de trabajo.

Se recomienda poner a prueba la capacidad de los servidores mediante prueba de estrés usando un grupo determinado de personas para acceder al mismo tiempo y comprobar la capacidad y eficiencia del servidor.

9. Anexos

9.1. Anexo 1: Entrevista Dir. Desarrollo

N° de Entrevista:	1	Fecha:	18 de septiembre del 2017
Punto de Reunión:	Sala de reuniones de DGSGI		

Entrevistado (s)	
Nombre	Cargo
Francisco Mora.	Responsables del Área de Desarrollo

Entrevistador (es)
Denis García

Tabla 7 Entrevista Dir. Desarrollo

Preguntas
Denis: Buenos días Francisco, ¿Qué podría decirme del sistema que vamos a construir?
Francisco: Buenos días Denis el sistema que se necesita es un sistema que nos sirva como herramienta para que las personas puedan llevar un autocontrol de las actividades que realizan, registrar los proyectos, las actividades y las tareas con el fin de para generar el plan operativo.
Denis: ¿Cómo se genera actualmente el plan operativo?
Francisco: Con Excel tenemos un archivo cada director de área, donde vamos llenándolo cronológicamente el orden de tareas y actividades dentro de un proyecto y de vez en cuando nos piden informes de actividades o personal que tenemos que sacar del mismo, el seguimiento a proyectos y actividades en realidad es muy débil.
Denis: ¿Y cuánto tiempo les toma aproximadamente la creación de este plan operativo?

Francisco: Esa pregunta es difícil porque a cómo te puede tomar unas 2 horas como queda sujeto a cambios debemos editarlo luego puede llegar a tomar días, pero por lo general te lo piden de un día para otro.

Denis: ¿Qué aspectos de la planificación operativa crees que necesita ser optimizado?

Francisco: Mira el seguimiento de actividades es algo muy necesario para lograr una mejor optimización del tiempo de desarrollo de los procesos. También tenemos que tener en cuenta los informes que se podrían desarrollar para obtener una información precisa y confiable, también nos gustaría implementar alguna herramienta o foro que agilice la comunicación, ya que es algo indispensable para cualquier equipo de trabajo y ayuda con este proceso de planificación, una de las variables que tenemos que tener muy en cuenta es el tiempo.

9.2. Anexo 2: Entrevista Dir. Geo repositorio

N° de Entrevista:	2	Fecha:	02 de octubre del 2017
Punto de Reunión:	Sala de reuniones de DGSGI		

Entrevistado (s)	
Nombre	Cargo
Meylin Sierra	Responsables del Área de <u>b</u> ases de datos

Entrevistador (es)
Denis García

Tabla 8 Entrevista Dir. Geo repositorio

Preguntas
<p>Meylin: Buen dia, ¿Qué es lo que necesitas para el sistema?</p> <p>Denis: Buenos días Meylin el sistema va a tener una base de datos de PosgreSql con un usuario de la aplicación para llevar un mejor, control también se va a implementar una tabla de auditoría que va a guardar acciones del usuario para mayor seguridad.</p>

Meylin: Tenemos varios servidores de bases de datos dependiendo de para lo que se necesite unos son de pruebas otros de producción y algunos con estándares geoespaciales que se ocupan para distintos sistemas,

Denis: ¿Que observaciones daría para que el sistema sea desarrollado de manera eficaz?

Meylin: Te voy a pasar el último archivo de plan operativo que tenemos es un Excel donde vas a ver como se definen las actividades y que es lo que se necesita para generar, con lo de llevar el control de bienes no vas a tener problema porque el SIAF ya lleva el control de los activos y pasivos de la institución.

Denis: ¿Que observaciones daría para el sistema?

Meylin: Pues que sea intuitivo, fácil de entender y adaptarse, los usuarios puedan llevar el control de su trabajo y poder validar el avance para llevar un mejor control.

9.3. Anexo 3: diseño de la base de datos

DbSchema
Default layout
Move the mouse over tables & columns to read the comments.

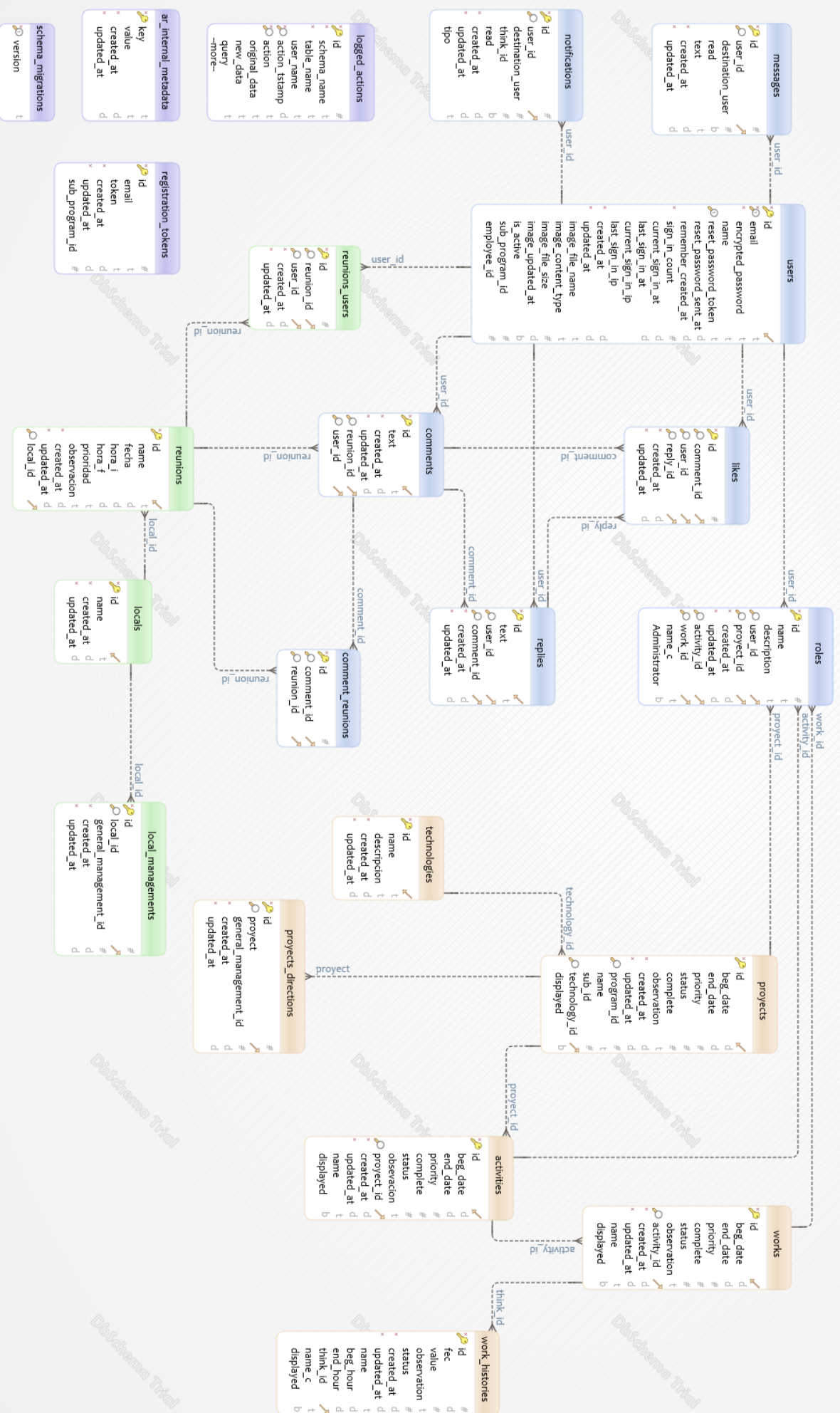


Figura 25 Diseño de base de datos

9.3.1. Tabla actividades

Tabla 9 Tabla actividades

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	beg_date	date	
	end_date	date	
	priority	integer	
	complete	integer	
	status	integer	
	obsevacion	text	
	proyect_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
	name	varchar	
	displayed	bool	
Indexes			
	activities_pkey	ON id	
	index_activities_on_proyect_id	ON proyect_id	
Foreign Keys			
	fk_rails_f200ba8dad	(proyect_id) ref proyects (id)	
Triggers			
	activity_if_modified_trg	CREATE TRIGGER activity_if_modified_tr	

Indexes	Field Name	Data Type	Description
		AFTER INSERT OR DELETE OR UPDATE ON activities FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.2. Tabla ar_internal_metadata

Tabla 10 Tabla ar_internal_metadata

Indexes	Field Name	Data Type	Description
*	key	varchar	
	value	varchar	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	ar_internal_metadata_pkey	ON key	

9.3.3. Tabla comment_reunions

Tabla 11 Tabla comment_reunions

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	comment_id	integer	
	reunion_id	integer	

Indexes	Field Name	Data Type	Description
Indexes			
	comment_reunions_pkey	ON id	
	index_comment_reunions_on_comment_id	ON comment_id	
	index_comment_reunions_on_reunion_id	ON reunion_id	
Foreign Keys			
	fk_rails_2003c64574	(comment_id) ref comments (id)	
	fk_rails_71883cebba	(reunion_id) ref reunions (id)	

9.3.4. Tabla comments

Tabla 12 Tabla comments

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	text	text	
*	created_at	timestamp	
*	updated_at	timestamp	
	reunion_id	integer	
	user_id	integer	
Indexes			
	comments_pkey	ON id	

Indexes	Field Name	Data Type	Description
	index_comments_on_reunion_id	ON reunion_id	
	index_comments_on_user_id	ON user_id	
Foreign Keys			
	fk_rails_80d333987f	(reunion_id) ref reunions (id)	
	fk_rails_03de2dc08c	(user_id) ref users (id)	

9.3.5. Tabla likes

Tabla 13 Tabla likes

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	comment_id	integer	
	user_id	integer	
	reply_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	likes_pkey	ON id	
	index_likes_on_comment_id	ON comment_id	
	index_likes_on_reply_id	ON reply_id	
	index_likes_on_user_id	ON user_id	

Indexes	Field Name	Data Type	Description
Foreign Keys			
	fk_rails_e92b21943f	(comment_id) ref comments (id)	
	fk_rails_8baaaa2c07	(reply_id) ref replies (id)	
	fk_rails_1e09b5dabf	(user_id) ref users (id)	

9.3.6. Tabla local_managements

Tabla 14 Tabla local_managements

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	local_id	integer	
	general_management_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	local_managements_pkey	ON id	
	index_local_managements_on_local_id	ON local_id	
Foreign Keys			

Indexes	Field Name	Data Type	Description
	fk_rails_72497928f6	(local_id) ref locals (id)	

9.3.7. Tabla locals

Tabla 15 Tabla locals

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	name	varchar	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	locals_pkey	ON id	

9.3.8. Tabla logged_actions

Tabla 16 Tabla logged_actions

Indexes	Field Name	Data Type	Description
*	id	bigserial AUTOINCREMENT	
*	schema_name	text	
*	table_name	text	

Indexes	Field Name	Data Type	Description
	user_name	text	
*	action_tstamp	timestampz DEFAULT now()	
*	action	text	
	original_data	text	
	new_data	text	
	query	text	
	((schema_name ' '::text) table_name))	varchar	
Indexes			
	logged_actions_pkey	ON id	
	logged_actions_action_idx	ON action	
	logged_actions_action_tstamp_idx	ON action_tstamp	
	logged_actions_schema_table_idx	ON ((schema_name ' '::text) table_name))	
Constraints			
	logged_actions_action_check	action = ANY (ARRAY['I '::text, 'D '::text, 'U '::text])	

9.3.9. Tabla messages

Tabla 17 Tabla messages

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	user_id	integer	
	destination_user	integer	
	read	bool	
	text	varchar	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	messages_pkey	ON id	
	index_messages_on_user_id	ON user_id	
Foreign Keys			
	fk_rails_273a25a7a6	(user_id) ref users (id)	
Triggers			
	messages_if_modified_trg	CREATE TRIGGER messages_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON messages FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.10. Tabla notifications

Tabla 18 Tabla notifications

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	user_id	integer	
	destination_user	integer	
	think_id	integer	
	read	bool	
*	created_at	timestamp	
*	updated_at	timestamp	
	tipo	varchar	
Indexes			
	notifications_pkey	ON id	
	index_notifications_on_user_id	ON user_id	
Foreign Keys			
	fk_rails_b080fb4855	(user_id) ref users (id)	

9.3.11. Tabla projects

Tabla 19 Tabla projects

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	

Indexes	Field Name	Data Type	Description
	beg_date	date	
	end_date	date	
	priority	integer	
	status	integer	
	complete	integer	
	observation	text	
*	created_at	timestamp	
*	updated_at	timestamp	
	program_id	integer	
	name	varchar	
	sub_id	integer	
	technology_id	integer	
	displayed	bool	
Indexes			
	projects_pkey	ON id	
	index_projects_on_program_id	ON program_id	
	index_projects_on_technology_id	ON technology_id	
Foreign Keys			
	fk_rails_51619aefb7	(technology_id) ref technologies (id)	

Indexes	Field Name	Data Type	Description
Triggers			
	projects_if_modified_trg	CREATE TRIGGER projects_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON projects FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.12. Tabla projects_directions

Tabla 20 Tabla projects_directions

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	project	integer	
	general_management_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	projects_directions_pkey	ON id	
	idx_projects_directions_project	ON project	
Foreign Keys			
	fk_projects_directions	(project) ref projects (id)	

Indexes	Field Name	Data Type	Description
Triggers			
	projects_dir_if_modified_trg	CREATE TRIGGER projects_dir_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON projects_directions FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.13. Tabla registration_tokens

Tabla 21 Tabla registration_tokens

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	email	varchar	
	token	varchar	
*	created_at	timestamp	
*	updated_at	timestamp	
	sub_program_id	integer	
Indexes			
	registration_tokens_pkey	ON id	

9.3.14. Tabla replies

Tabla 22 Tabla replies

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	text	text	
	user_id	integer	
	comment_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	replies_pkey	ON id	
	index_replies_on_comment_id	ON comment_id	
	index_replies_on_user_id	ON user_id	
Foreign Keys			
	fk_rails_f13795d06e	(comment_id) ref comments (id)	
	fk_rails_256e4b72c5	(user_id) ref users (id)	

9.3.15. Tabla reunions

Tabla 23 Tabla reunions

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	name	varchar	

Indexes	Field Name	Data Type	Description
	fecha	date	
	hora_i	time	
	hora_f	time	
	prioridad	varchar	
	observacion	text	
*	created_at	timestamp	
*	updated_at	timestamp	
	local_id	integer	
Indexes			
	reunions_pkey	ON id	
	index_reunions_on_local_id	ON local_id	
Foreign Keys			
	fk_rails_703cd18f2b	(local_id) ref locals (id)	
Triggers			
	reunions_if_modified_trg	CREATE TRIGGER reunions_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON reuniones FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.16. Tabla reunions_users

Tabla 24 Tabla reunions_users

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	reunion_id	integer	
	user_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	reunions_users_pkey	ON id	
	idx_reunions_users_reunion_id	ON reunion_id	
	idx_reunions_users_user_id	ON user_id	
Foreign Keys			
	fk_reunions_users_reunions	(reunion_id) ref reunions (id)	
	fk_reunions_users_users	(user_id) ref users (id)	

9.3.17. Tabla roles

Tabla 25 Tabla roles

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	name	varchar	

Indexes	Field Name	Data Type	Description
	description	text	
	user_id	integer	
	proyect_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
	activity_id	integer	
	work_id	integer	
	name_c	varchar	
	Administrator	bool	
Indexes			
	roles_pkey	ON id	
	index_roles_on_activity_id	ON activity_id	
	index_roles_on_proyect_id	ON proyect_id	
	index_roles_on_user_id	ON user_id	
	index_roles_on_work_id	ON work_id	
Foreign Keys			
	fk_rails_954ae45047	(activity_id) ref activities (id)	
	fk_rails_fc77300b34	(proyect_id) ref proyects (id)	
	fk_rails_ab35d699f0	(user_id) ref users (id)	
	fk_rails_98f708517a	(work_id) ref works (id)	
Triggers			
	roles_if_modified_trg	CREATE TRIGGER roles_if_modified_trg AFTER	

Indexes	Field Name	Data Type	Description
		INSERT OR DELETE OR UPDATE ON roles FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.18. Tabla schema_migrations

Tabla 26 Tabla schema_migrations

Indexes	Field Name	Data Type	Description
*	version	varchar	
Indexes			
	unique_schema_migrations	ON version	

9.3.19. Tabla technologies

Tabla 27 Tabla technologies

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	name	varchar	
	descripcion	text	
*	created_at	timestamp	
*	updated_at	timestamp	
Indexes			
	technologies_pkey	ON id	
Triggers			

Indexes	Field Name	Data Type	Description
	technologies_if_modified_trg	CREATE TRIGGER technologies_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON technologies FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.20. Tabla users

Tabla 28 Tabla users

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
*	email	varchar DEFAULT "::character varying	
*	encrypted_password	varchar DEFAULT "::character varying	
	name	varchar	
	reset_password_token	varchar	
	reset_password_sent_at	timestamp	
	remember_created_at	timestamp	
*	sign_in_count	integer DEFAULT 0	
	current_sign_in_at	timestamp	
	last_sign_in_at	timestamp	
	current_sign_in_ip	inet	

Indexes	Field Name	Data Type	Description
	last_sign_in_ip	inet	
*	created_at	timestamp	
*	updated_at	timestamp	
	image_file_name	varchar	
	image_content_type	varchar	
	image_file_size	integer	
	image_updated_at	timestamp	
	is_active	bool	
	sub_program_id	integer	
	employee_id	integer	
Indexes			
	users_pkey	ON id	
	index_users_on_email	ON email	
	index_users_on_reset_password_token	ON reset_password_token	
Triggers			
	users_if_modified_trg	CREATE TRIGGER users_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON users FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.21. Tabla work_histories

Tabla 29 Tabla work_histories

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	fec	date	
	value	integer	
	observation	text	
	status	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
	name	varchar	
	beg_hour	time	
	end_hour	time	
	think_id	integer	
	name_c	varchar	
	displayed	bool	
Indexes			
	work_histories_pkey	ON id	
Foreign Keys			
	work_histories_works_fkey	(think_id) ref works (id)	
Triggers			
	work_histories_if_modified_trg	CREATE TRIGGER work_histories_if_modified_trg AFTER INSERT OR DELETE	

Indexes	Field Name	Data Type	Description
		OR UPDATE ON work_histories FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

9.3.22. Tabla works

Tabla 30 Tabla works

Indexes	Field Name	Data Type	Description
*	id	serial AUTOINCREMENT	
	beg_date	date	
	end_date	date	
	priority	integer	
	complete	integer	
	status	integer	
	observation	text	
	activity_id	integer	
*	created_at	timestamp	
*	updated_at	timestamp	
	name	varchar	
	displayed	bool	
Indexes			
	works_pkey	ON id	
	index_works_on_activity_id	ON activity_id	

Indexes	Field Name	Data Type	Description
Foreign Keys			
	fk_rails_63ba084913	(activity_id) ref activities (id)	
Triggers			
	works_if_modified_trg	CREATE TRIGGER works_if_modified_trg AFTER INSERT OR DELETE OR UPDATE ON works FOR EACH ROW EXECUTE PROCEDURE if_modified_func()	

Powered by DbSchema

9.4. Anexo 4: Cronograma de ejecución

9.5. Anexo 5: Ficha de registro de prueba

Registro de Pruebas		
Indicaciones: Ejecute cada prueba rellorando los valores que a continuación se le piden, primero en una hoja de Excel de la manera tradicional y luego con las herramientas del sistema hoja de tiempo. En cada prueba anote los datos que se solicitan: <ul style="list-style-type: none"> • Tiempo: hora de inicio y fin de la prueba (con minutos y segundos) • Pasos: Conteo de la cantidad de pasos que requieren para generar el reporte. • Simplicidad: Percepción que tenga de la facilidad de uso del servidor de reportes, es simple o complejo. Considere si la interfaz de usuario está bien diseñada, es clara, fácil de entender, intuitiva, y los controles usados son conocidos y de fácil uso. 		
Test1: Creación de estructura de trabajo (al menos 10 actividades y tareas).		
Medida:	Hoja Excel	Sistema hoja de tiempo
Tiempo:		
Hora de Inicio:		
Hora finalización:		
Pasos de procedimiento:		
Simplicidad de uso:	[] Simple, [] Normal, [] Complejo	[] Simple, [] Normal, [] Complejo
Test2: Creación de reporte de plan operativo anual		
Medida:	Hoja Excel	Sistema hoja de tiempo
Tiempo:		
Hora de Inicio:		
Hora finalización:		
Pasos de procedimiento:		
Simplicidad de uso:	[] Simple, [] Normal, [] Complejo	[] Simple, [] Normal, [] Complejo
Test3: Validación de resultados (feedback)		
Medida:	Hoja Excel	Sistema hoja de tiempo
Tiempo:		
Hora de Inicio:		

Hora finalización:		
Pasos de procedimiento:		
Simplicidad de uso:	[<input type="checkbox"/>] Simple, [<input type="checkbox"/>] Normal, [<input type="checkbox"/>] Complejo	[<input type="checkbox"/>] Simple, [<input type="checkbox"/>] Normal, [<input type="checkbox"/>] Complejo

9.6. Anexo 6: Registro de reuniones


Registro de reuniones				
Las reuniones que se hicieron a lo largo del ciclo de vida de este proyecto han sido después de cada iteración con las cuales se entregan resultados y se revisan aspectos de todo tipo para verificar que las características sean las adecuadas, aquí también es donde se registra la retroalimentación por parte de los usuarios del sistema para próximas iteraciones.				
Reunión 1: Revisiones de casos de uso				
Hora inicio:	09:05am	Día:	05-02-2018	
Hora fin:	11:30am	Importancia:	Medio	
Integrantes:	<ul style="list-style-type: none">Francisco MoraBinnia PadillaDenis GarcíaFabiola Castillo			
Observaciones:	<ul style="list-style-type: none">Se verificaron que los casos de uso cumplan con los requisitos mínimos del sistemaSe observó también algunas restricciones de permisos que necesitan ser cambiadas en cuanto a navegación y edición de tareas			
Entregas:	<ul style="list-style-type: none">Casos de uso primera versiónVistas de proyectos, actividades y tareas			
Reunión 2: verificación de cambios de iteración				
Hora inicio:	01:05pm	Día:	14-03-2018	
Hora fin:	02:20pm	Importancia:	Alto	


Integrantes:	<ul style="list-style-type: none">• Francisco Mora• Denis García• Fabiola Castillo• Vladimir Gutiérrez			
Observaciones:	<ul style="list-style-type: none">• Se validaron las vistas propuestas• Se establecieron algunos gráficos para ser mostrados• Revisión de la versión			
Entregas	<ul style="list-style-type: none">• Casos de uso nueva versión• Nueva versión de vistas			
Reunión 3: Versión 1 aplicación				
Hora inicio:	01:05pm	Día:	16-04-2018	
Hora fin:	02:00pm	Importancia:	Alto	
Integrantes:	<ul style="list-style-type: none">• Francisco Mora• Denis García			
Observaciones:	<ul style="list-style-type: none">• Se definieron algunas vistas de reportes necesarias• Se estableció el rol administrador para administrar los puntos delicados del sistema como usuarios• Verificación de funcionalidades• Revisión de la versión			
Entregas	<ul style="list-style-type: none">• Versión de la aplicación			
Reunión 4: Revisión de aplicación				
Hora inicio:	08:25am	Día:	09-05-2018	
Hora fin:	11:30am	Importancia:	medio	
Integrantes:	<ul style="list-style-type: none">• Fabiola Castillo• Denis García			
Observaciones:	<ul style="list-style-type: none">• Se definieron ajustes para el calendario• Se estableció crear un calendario para visualizar las tareas y reuniones• Revisión de la versión			

Reunión 5: versión 2 aplicación				
Hora inicio:	01:25pm	Día:	05-06-2018	
Hora fin:	03:30pm	Importancia:	alto	
Integrantes:	<ul style="list-style-type: none"> • Fabiola Castillo • Denis García • Luis Herrera • Vladimir Gutiérrez 			
Observaciones:	<ul style="list-style-type: none"> • Se definieron nuevas herramientas como un organigrama, mensajes y estrellas en validación de avances de tareas, y el autoseguimiento. 			
Entregas:	<ul style="list-style-type: none"> • Aplicación versión 2 • Calendario, chat app, mejora en la creación de estructura. 			

9.7. Anexo 7: Versiones de aplicación

9.7.1. Anexo 7.1: vista de tareas versión 1.1


**INETER**
Instituto Nicaragüense de Estudios Territoriales

3

Inicio

Actividades

Tareas



Tareas

Proyectos

Evolución

Mitrab

MARENA

AEEI

Tripe A

Sistema MARENA

INSS


Nombre	Fecha inicio	Fecha fin	Progreso	Asignado a
identificar los recursos	2017-02-08	2017-02-22	100%	Kevin
identificar los resultados	2017-02-02	2017-02-16	100%	Kevin
realizar las vistas	2017-02-16	2017-02-24	0%	Kevin


revisar el protocolo


revisar el protocolo

revisar el protocolo


1 2 > >>

Denis te ha asignado un/a actividad




Denis te ha asignado un/a proyecto




Kevin te ha asignado un/a proyecto




por nombre



ción Actividad










localhost:3000/works#


Figura 37 vista tareas v1.1

9.7.2. Anexo 7.2: vista de lista de proyectos versión 2.1












INETER
 Instituto Nicaragüense de Estudios Territoriales


 Inicio
 Actividades
 Tareas
 

Proyectos DGSB



1 - Apoyo Institucional
 2 - Hoja de Tiempo
 3 - Reuniones- DGSB
 4 - Seguimiento DGSB
 5 - Sistema en Línea INETER

Nombre	Inicio	Fin	Prioridad	Estado	Completado	Observación	
Plataforma de interoperabilidad para auxiliar con la gestión hidrometeorológica de la cuenca del Río Dipilto	21/08/2017	30/06/2018	Alta	En proceso	0	Consultoría	
Capacitaciones	01/06/2017	15/12/2017	Baja	En proceso	50	Cursos Inglés INATEC. 7am - 9am	
Seguimiento administrativo OAIP	15/06/2017	22/12/2017	Medio	En proceso	0		
Fortalecimiento Educativo	13/06/2017	13/06/2019	Baja	En proceso	0	Curso de INGLES	
Informes	16/06/2017	22/12/2017	Baja	En proceso	0		
Oficina de Ética	12/06/2017	22/12/2017	Medio	En proceso	0		
Inventario	14/08/2017	15/09/2017	Baja	En proceso	0		
Auditorio	12/06/2017	22/12/2017	Baja	En proceso	0		
Solvencias	12/06/2017	22/12/2017	Medio	En proceso	0		
Base de datos bibliograficos	12/06/2017	22/12/2017	Alta	En proceso	0		

1
 2
 3
 4
 5
 >
 »

CREAR PROYECTO
 INFORME
 GRÁFICOS

© 2017 Copyright Ineter
 Area de trabajo

Figura 38 vista listar proyectos v2.1

9.7.3. Anexo 7.3: vista calendario versión 2.1

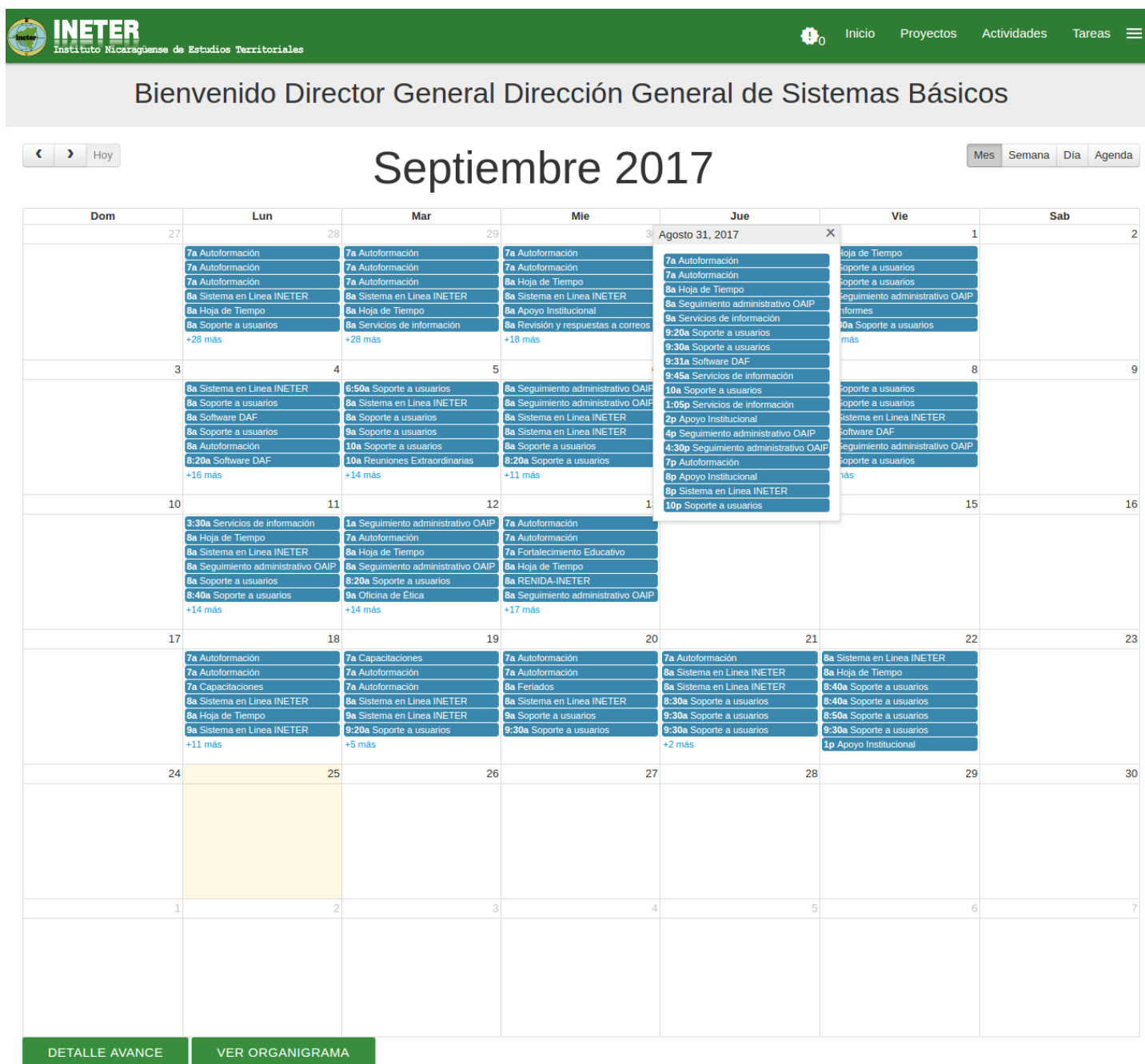


Figura 39 vista calendario v2.1

9.7.4. Anexo 7.4: formulario editar proyecto versión 2.1

INETER
 Instituto Nicaragüense de Estudios Territoriales

Inicio
 Actividades
 Tareas

Editar proyecto

Nombre

Hoja de Tiempo

Fecha inicio
16-01-2017

Fecha finalización
31-03-2017

Prioridad
Baja

Estado
En proceso

Observación

Dirección
DGSB

Sub dirección
D.D

ATRÁS

GUARDAR

© 2017 Copyright Ineter

Area de trabajo

Figura 40 formulario editar proyecto v2.1

9.7.5. Anexo 7.5: vista mostrar proyecto

INETER
 Instituto Nicaragüense de Estudios Territoriales

Inicio
 Actividades
 Tareas

Proyectos

- 1 - Apoyo Institucional
- 2 - Hoja de Tiempo
- 3 - Reuniones- DGSB
- 4 - Seguimiento DGSB
- 5 - Sistema en Linea INETER

Proyecto

2017 > DGSB > Sistema en Linea ...

PDF EDITAR

Inicia: 02/05/2017 Finaliza: 15/12/2017
 Prioridad: Alta Sub dirección: D.D Observacion:

ATRÁS

AGREGAR USUARIO

AVANCE

NUEVA ACTIVIDAD

23 %

Actividades

Nombre	Recursos Humanos	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Observación																																							
Servicios en Linea	F I F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
PREPARACION AMBIENTE DESARROLLO	W B c K F M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
Tramites en linea 18 Permisos Marena	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
ANALISIS Y DISEÑO	K c S D J D F G I B F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
PRUEBAS FUNCIONALES	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
CAPACITACION USUARIOS x TRAMITE	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
PREPARACION AMBIENTE DE PRODUCCION	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
DESARROLLO	S K J D c F F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
INSTALACION Y CONFIGURACION DE EQUIPOS DE SEGURIDAD	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
DOCUMENTACION DEL SISTEMA	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
IMPLEMENTACION	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
Plan de Migración de proyectos TFS	F F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
Documentación del Sistema	F I F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52

© 2017 Copyright Ineter

Area de trabajo

Figura 41 vista mostrar proyecto v2.1

9.7.6. Anexo 7.6: vista de lista de proyectos versión 2.2

Proyectos

Proyectos Departamento de Desarrollo

Buscar por nombre

Nombre	Inicio	Fin	Estado	Prioridad	Observación	Opciones
Plataforma de interoperabilidad para auxiliar con la gestión hidrometeorológica de la cuenca del Río Dipilto	21/08/2017	30/06/2018	En proceso	Alta	Consultoría	[Icono de play]
Capacitaciones	01/06/2017	15/12/2017	En proceso	Baja	Cursos Inglés INATEC. 7am - 9am	[Icono de play]
Seguimiento DGSB	01/05/2017	31/12/2017	En proceso	Baja	Por orientación de la CO-Dirección. Seguimiento a las actividades de Desarrollo e IT.	[Icono de play], [Icono de edit], [Icono de delete]
Gestiones Administrativas	01/05/2017	22/12/2017	En proceso	Medio	Gestiones de Caja Chica, Solicitud de artículos de almacén, Almacén, etc.	[Icono de play]
Feriatos	01/06/2017	31/12/2017	En proceso	Baja	Días feriados	[Icono de play]
Acompañamiento Catastral	01/05/2017	31/12/2017	En proceso	Baja		[Icono de play]
Lotes Ende-Red	24/04/2017	30/06/2017	En proceso	Baja		[Icono de play]
Prácticas Profesionales	17/04/2017	31/08/2017	Detenido	Baja	Prácticas profesionales 400 Horas.	[Icono de play]

1 2 > >>

CREAR PROYECTO INFORME GRÁFICOS HISTORIAL

© 2017 Copyright Ineter Area de trabajo

Figura 42 vista lista de proyectos v2.2

9.7.7. Anexo 7.7: vista de mostrar proyecto versión 2.2

Proyecto

2017 > DGSB > Hoja de Tiempo

Inicia: 16/01/2017 Finaliza: 31/03/2017 Prioridad: Baja Sub dirección: Departamento de Desarrollo

Observación:

ATRÁS PDF EDITAR AGREGAR USUARIO NUEVA ACTIVIDAD

86 %

Actividades

Nombre	Recursos Humanos	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Observación
Implementación del Sistema	I F	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52												
Desarrollo de sistema	D	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52												
Pruebas al sistema	D I	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52												
Documentación del Sistema	I F D	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52												

© 2017 Copyright Ineter Area de trabajo

Figura 43 vista de mostrar proyecto v2.2

9.7.8. Anexo 7.8: vista de reuniones versión 2.2

INETER

Instituto Nicaragüense de Estudios Territoriales

Inicio

Proyectos

Actividades

Tareas

Reuniones

Nombre	fecha	Hora Inicio	Hora Fin	Usuarios	Observación	
Reunión Co-Director. Presentación formal Asesor Técnico	11/07/2017	09:00 AM	11:00 AM	<div><div><div>Francisco Mora</div><div>Jeysson Gonzalez</div><div>Suyen Morales</div><div>Dusstin</div><div>Josseling Vargas</div><div>Dolores Rodriguez Ponce</div><div>Binia Padilla</div></div><div><div>Wesley Sang</div><div>Kevin Gaitan</div><div>Denis Guevara</div><div>Omar Medina Bermudez</div><div>Nicole Brenes</div><div>Ernesto</div><div>Meyling Sierra</div></div><div><div>Fabiola Castillo</div><div>cristhianxavier</div><div>Iván Urbina</div><div>Bernardo Orozco</div><div>Gerald Guevara</div><div>Meyling Sierra</div></div></div>	Presentación Asesor Técnico. Otros Temas	<div><div></div><div></div></div>
Reunión diagrama de objetos	03/07/2017	13:00 PM	14:30 PM	<div><div><div>cristhianxavier</div><div>Jeysson Gonzalez</div></div><div><div>Suyen Morales</div><div>Denis Guevara</div></div></div>	se analizaron las posibles afectaciones a atributos de algunas de las entidades con referencia a la nueva forma de pago	<div><div></div><div></div></div>
SLI estructura DDD	29/06/2017	16:00 PM	17:00 PM	<div><div><div>Jeysson Gonzalez</div><div>Suyen Morales</div></div><div><div>cristhianxavier</div><div>Kevin Gaitan</div><div>Denis Guevara</div></div></div>	Se demostró tracking de sistemas en línea para la estructura DDD	<div><div></div><div></div></div>
Acto natalicio de Carlos Fonseca	23/06/2017	08:30 AM	10:00 AM	<div><div><div>Jeysson Gonzalez</div><div>Suyen Morales</div><div>Francisco Mora</div><div>Denis Guevara</div></div><div><div>Dusstin</div><div>Kevin Gaitan</div><div>Gerald Guevara</div></div><div><div>cristhianxavier</div><div>Iván Urbina</div><div>Binia Padilla</div></div></div>	Acto en el parqueo de la DISUP por el natalicio de Carlos Fonseca	<div><div></div><div></div></div>
Presentacion Visual Studio	09/06/2017	09:00 AM	11:00 AM	<div><div><div>Francisco Mora</div><div>Jeysson Gonzalez</div></div><div><div>cristhianxavier</div><div>Denis Guevara</div><div>Meyling Sierra</div></div><div><div>Suyen Morales</div></div></div>	Presentación por parte del Representante de Microsoft Zev Yanovich	<div><div></div><div></div></div>
Equipo Desarrollo	01/06/2017	09:30 AM	10:00 AM	<div><div><div>Kevin Gaitan</div><div>Jeysson Gonzalez</div><div>Francisco Mora</div></div><div><div>Suyen Morales</div><div>Denis Guevara</div></div><div><div>cristhianxavier</div></div></div>	Definición de actividades de desarrollo en el proyecto TeamFoundation del Sistema en Línea	<div><div></div><div></div></div>
Reunión Equipo Desarrollo	23/05/2017	16:00 PM	17:00 PM	<div><div><div>Kevin Gaitan</div><div>Denis Guevara</div></div><div><div>Suyen Morales</div><div>cristhianxavier</div></div><div><div>Jeysson Gonzalez</div><div>Francisco Mora</div></div></div>	Lista de Tecnologías a utilizar en el desarrollo del proyecto. Creación de proyecto en el TSF. Trabajo en Front-End y Back-End. Recordar buenas prácticas en el desarrollo. definir arquitectura del proyecto de desarrollo en Visual Studio.	<div><div></div><div></div></div>
Actividades Natalicio Sandino	18/05/2017	13:00 PM	17:00 PM	<div><div><div>Jeysson Gonzalez</div><div>Fabiola Castillo</div><div>Dusstin</div><div>Francisco Mora</div><div>Josseling Vargas</div><div>Dolores Rodriguez Ponce</div><div>Wesley Sang</div></div><div><div>Kevin Gaitan</div><div>Iván Urbina</div><div>Omar Medina Bermudez</div><div>Gerald Guevara</div><div>Nicole Brenes</div><div>Bernardo Orozco</div><div>Meyling Sierra</div></div><div><div>cristhianxavier</div><div>Denis Guevara</div><div>Binia Padilla</div><div>Ernesto</div><div>Meyling Sierra</div></div></div>	Celebración del Natalicio de Sandino. La DGSB participó en la Liga Relámpago de Football	<div><div></div><div></div></div>
Reunión Co-Director INETER	15/05/2017	14:00 PM	16:30 PM	<div><div><div>Jeysson Gonzalez</div><div>Suyen Morales</div><div>Dusstin</div><div>Gerald Guevara</div><div>Meyling Sierra</div></div><div><div>Kevin Gaitan</div><div>Denis Guevara</div><div>Omar Medina Bermudez</div><div>Fabiola Castillo</div><div>Meyling Sierra</div></div><div><div>cristhianxavier</div><div>Iván Urbina</div><div>Binia Padilla</div><div>Francisco Mora</div></div></div>	Reunión solicitada por Co-Director INETER	<div><div></div><div></div></div>
Reunión para plan de trabajo de los trámites en Línea por	15/05/2017	10:00 AM	11:00 AM	<div><div><div>Jeysson Gonzalez</div><div>Suyen Morales</div><div>Dusstin</div><div>Gerald Guevara</div></div><div><div>Kevin Gaitan</div><div>Denis Guevara</div><div>Iván Urbina</div><div>Binia Padilla</div></div><div><div>cristhianxavier</div><div>Francisco Mora</div><div>Omar Medina Bermudez</div><div>Fabiola Castillo</div></div></div>	Presentar el nuevo plan de los trámites en línea para Desarrollo y BD	<div><div></div><div></div></div>

Figura 44 vista de reuniones v2.2

9.7.9. Anexo 7.9: vista de calendario versión 3.1

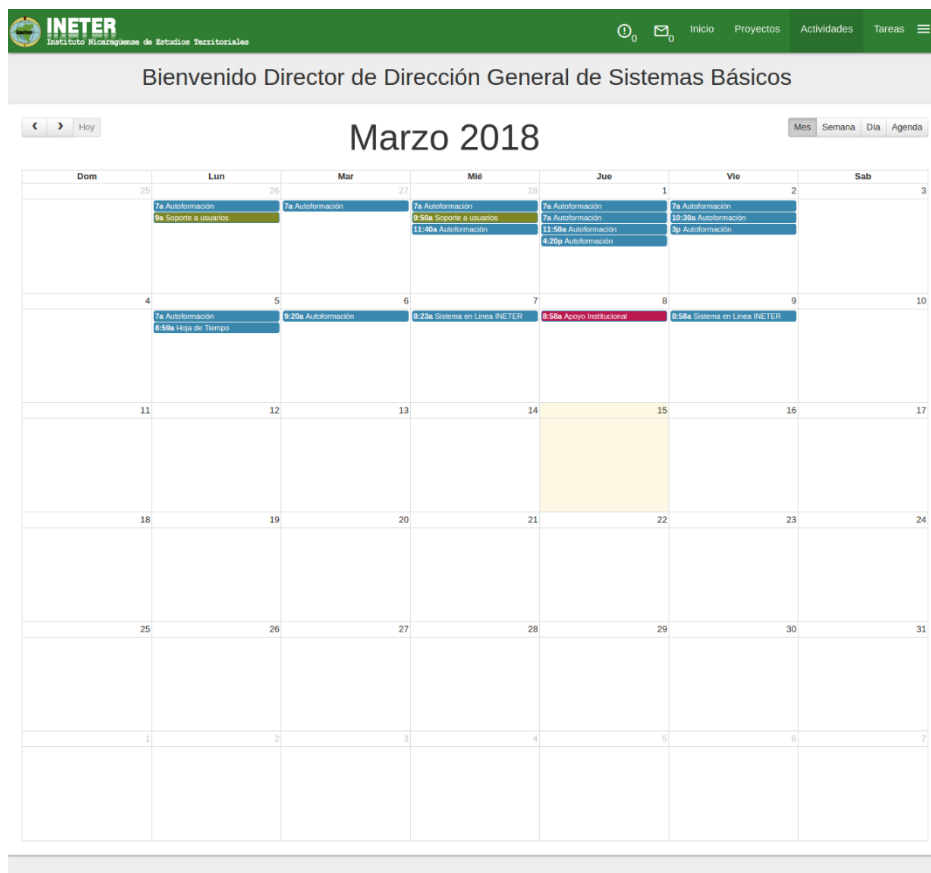


Figura 45 vista calendario v3.1

9.7.10. Anexo 7.10: vista de organigrama versión 3.1

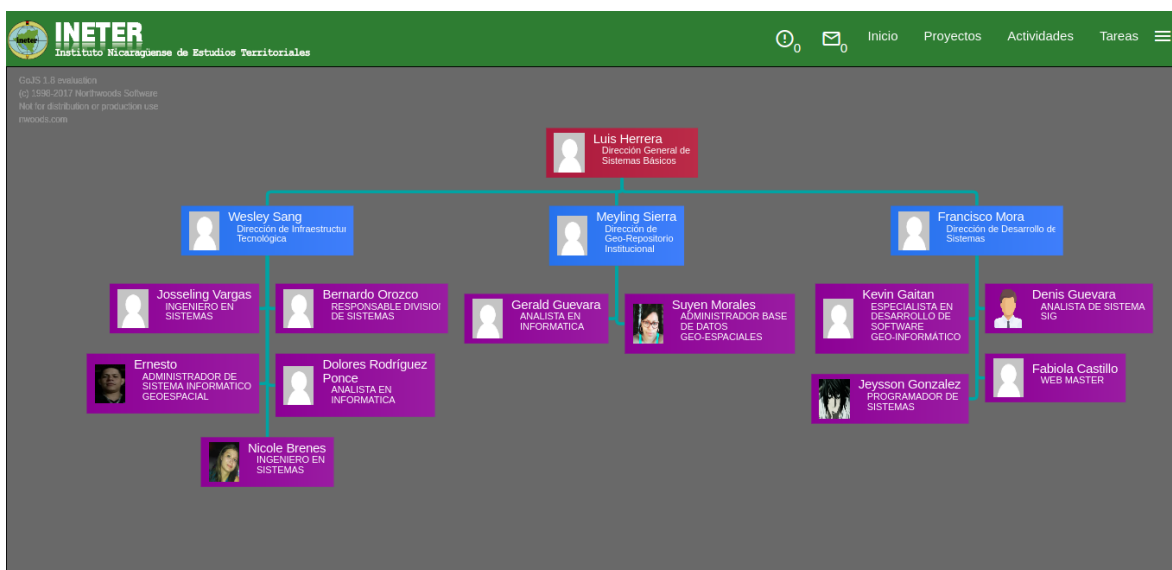


Figura 46 vista de organigrama v3.1

9.7.11. Anexo 7.11: vista de mostrar proyecto versión 3.1

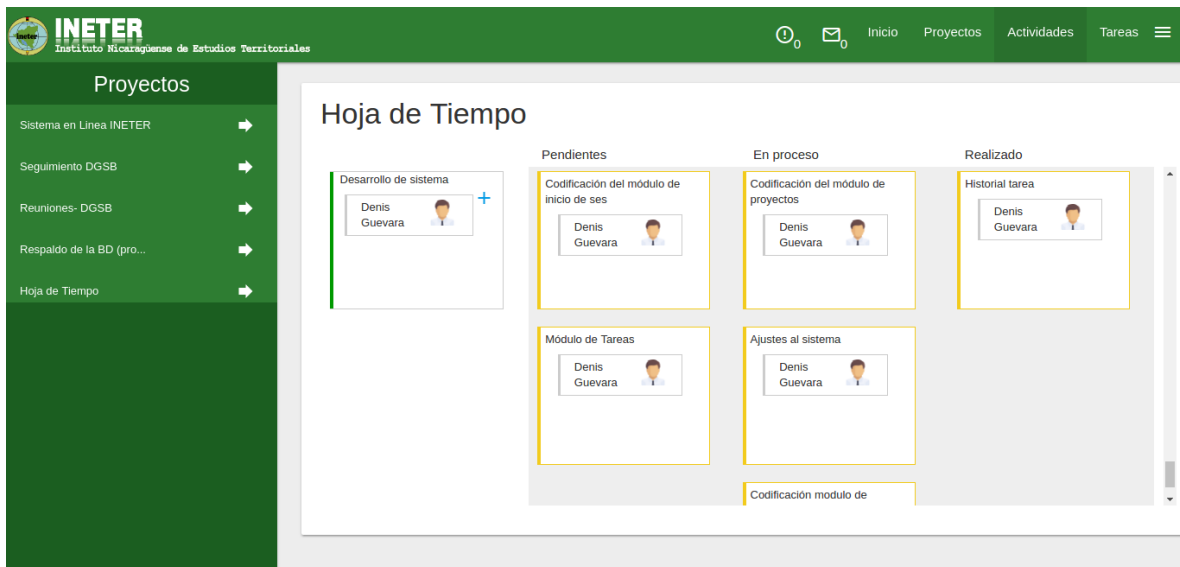


Figura 47 vista mostrar proyecto v3.1

9.7.12. Anexo 7.12: vista últimos avances registrados versión 3.2

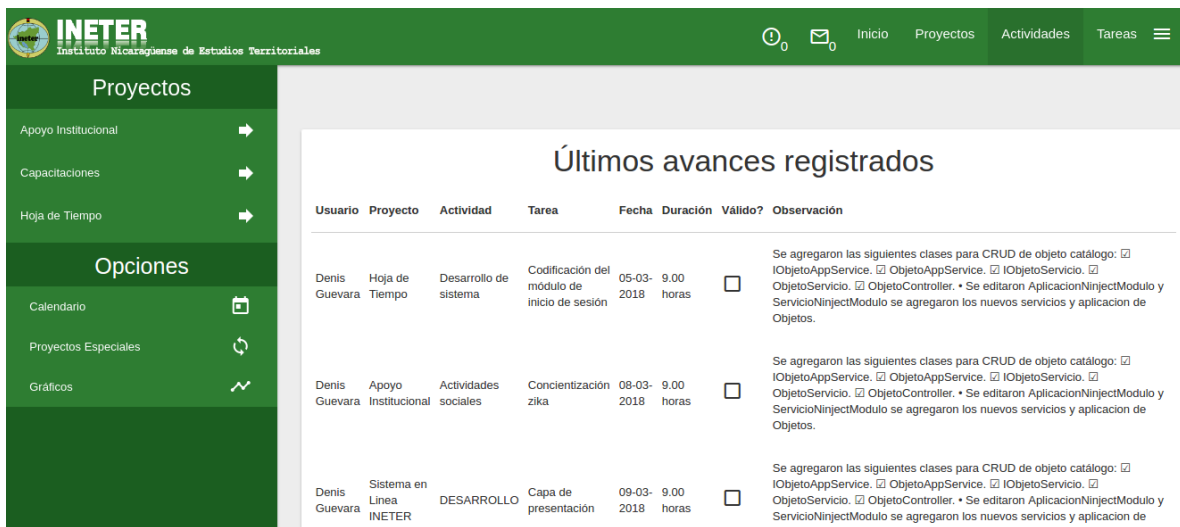
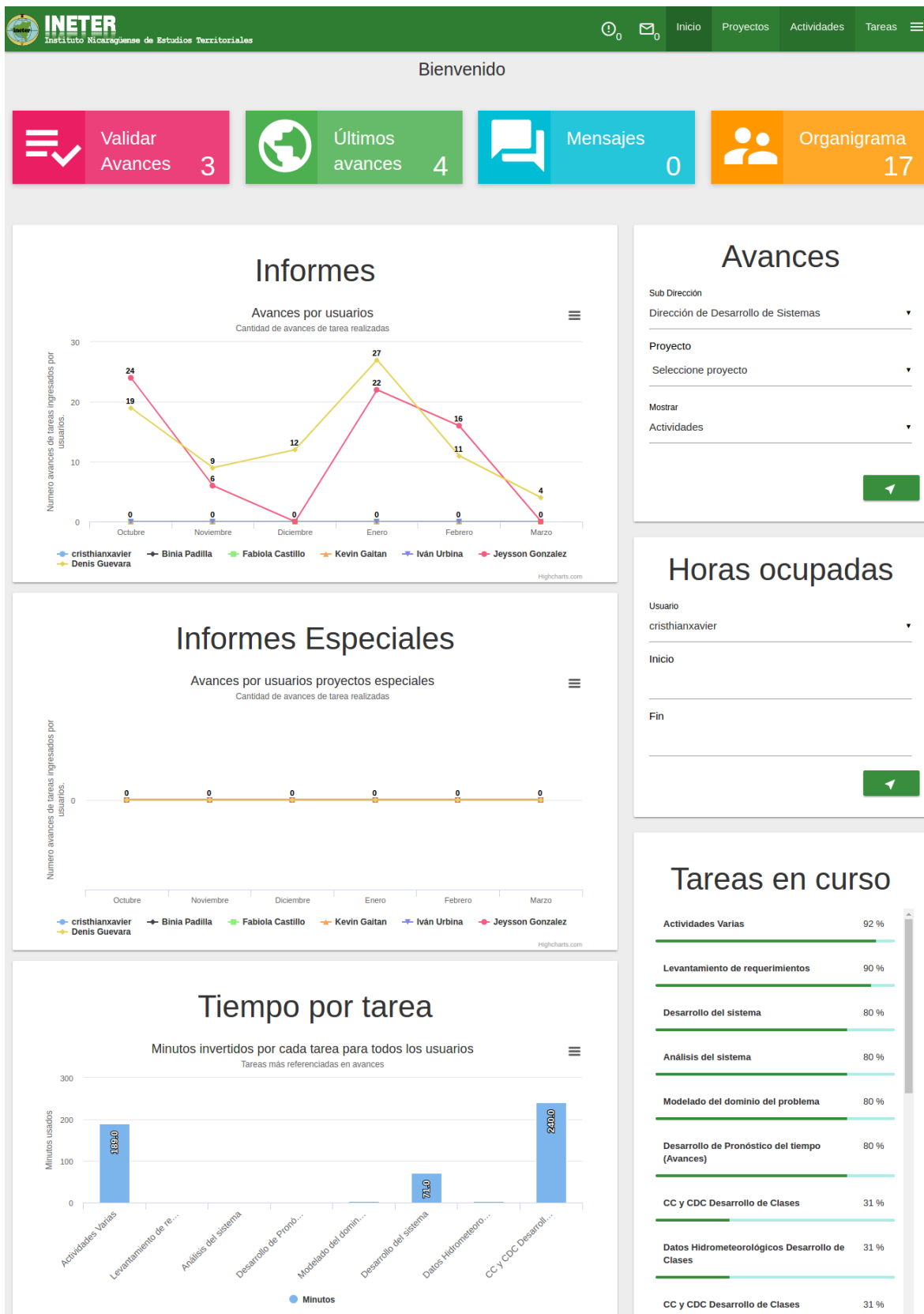


Figura 48 últimos avances registrados v3.2

9.7.13. Anexo 7.13: vista de informe





Estados:

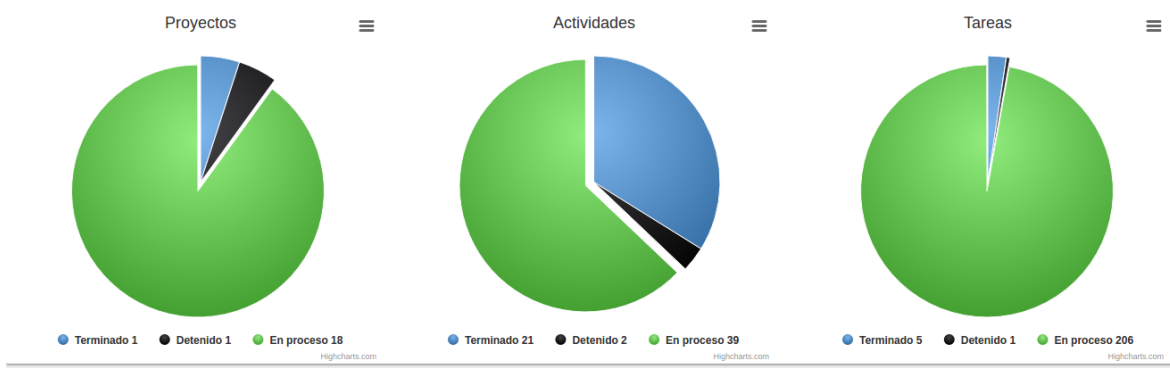


Figura 49 vista de informes v3.2

9.7.14. Anexo 7.14: vista validar tareas versión 3.2

INETER
Instituto Nicaragüense de Estudios Territoriales

Inicio Proyectos Actividades Tareas

Detalle de avance por usuario

Filtrar por usuario
Denis Guevara

Últimos registros

Fecha	Tarea	Estado
Lunes 05/03/2018	Hoja de Tiempo >Desarrollo de sistema >>Codificación del módulo de inicio de sesión De: 8:59:00am A: 5:59:00pm	✓
Jueves 08/03/2018	Apoyo Institucional >Actividades sociales >>Concientización Zika De: 8:58:00am A: 5:58:00pm	✓
Viernes 09/03/2018	Sistema en Línea INETER >DESARROLLO >>Capa de presentación De: 8:58:00am A: 5:58:00pm	✓

Figura 50 vista filtrar avance de tarea v3.2

9.7.15. Anexo 7.15: vista invalidar tarea versión 3.2

INETER
Instituto Nicaragüense de Estudios Territoriales

Inicio Proyectos Actividades Tareas

Detalle de avance

Viernes
09/03/2018

Sistema en Línea INETER
>DESARROLLO
>>Capa de presentación

Se agregaron las siguientes clases para CRUD de objeto catálogo: ☒ IObjetoAppService, ☒ ObjetoAppService, ☒ IObjetoServicio, ☒ ObjetoServicio, ☒ ObjetoController. • Se editaron AplicacionNinjectModulo y ServicioNinjectModulo se agregaron los nuevos servicios y aplicacion de Objetos.

De: 8:58:00am
A: 5:58:00pm

☐ Valido?

Por favor agregue un mensaje

Mensaje de validación por que esta mal la entrada o que cambia

CERRAR ENVIAR

Figura 51 vista invalidar tarea v3.2

9.7.16. Anexo 7.16: vista validar tarea versión 3.2

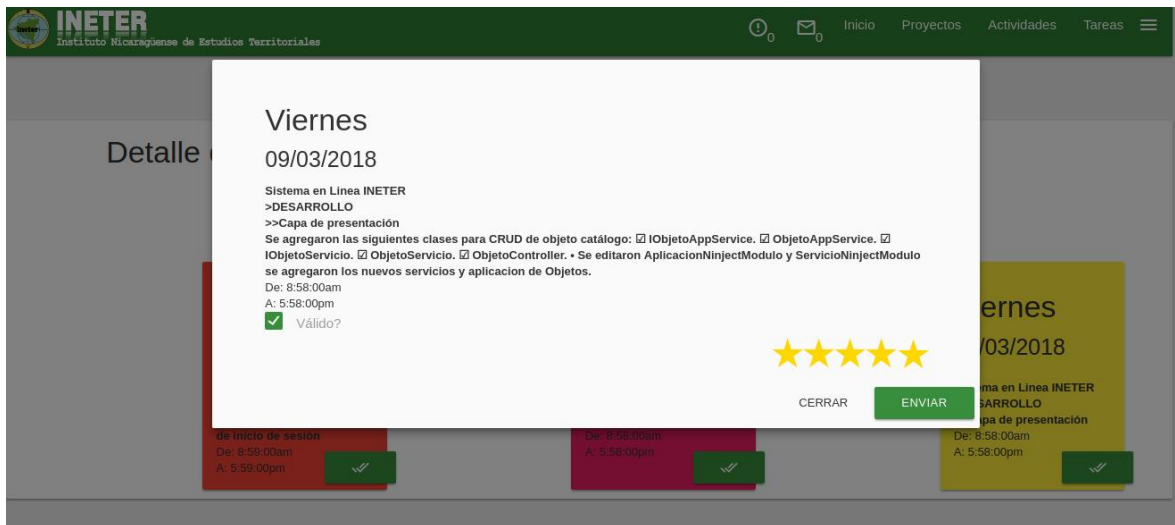


Figura 52 vista validar tarea v3.2

9.7.17. Anexo 7.17: vista de mensajes versión 3.3

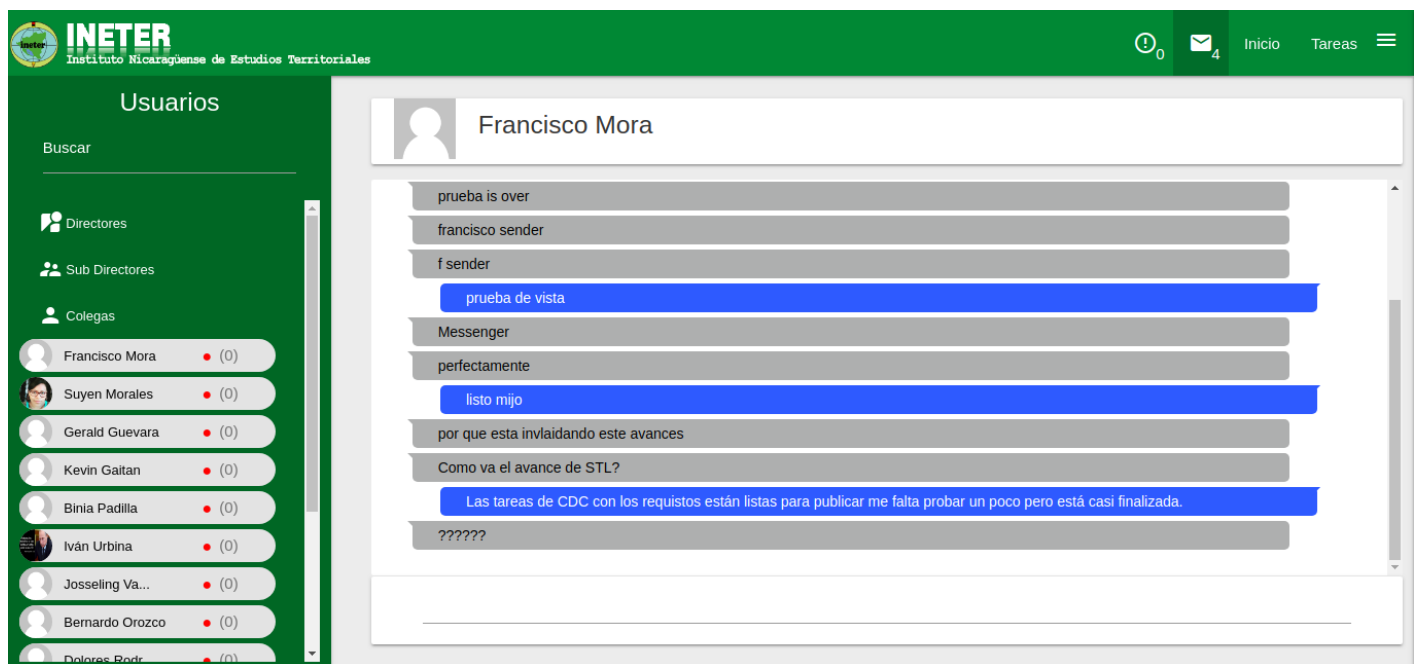


Figura 53 vista de mensajes v3.3

9.7.18. Anexo 7.18: vista de calendario versión 3.3

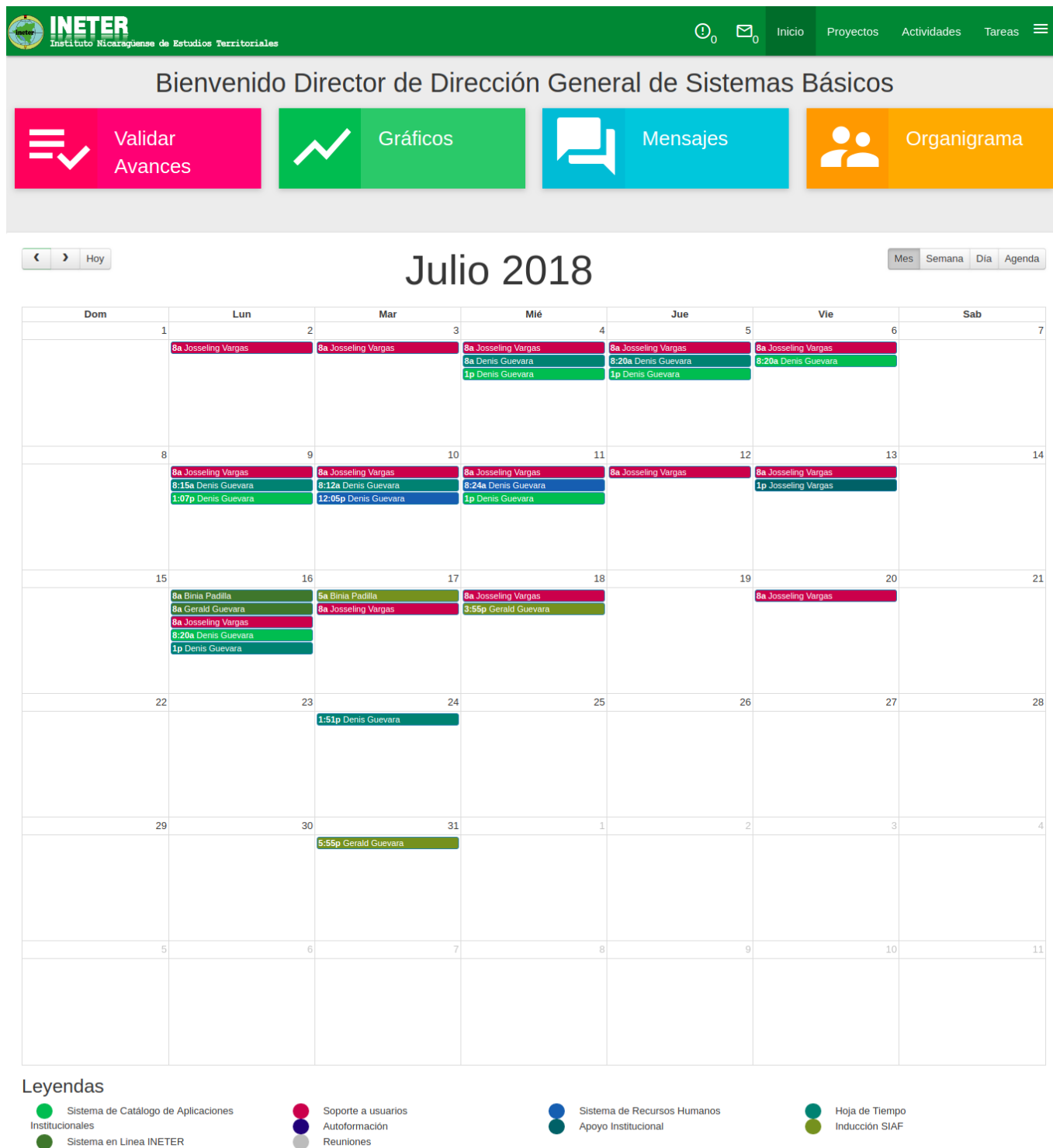


Figura 54 vista calendario v3.3

9.7.19. Anexo 7.19: vista mostrar proyecto versión 3.3

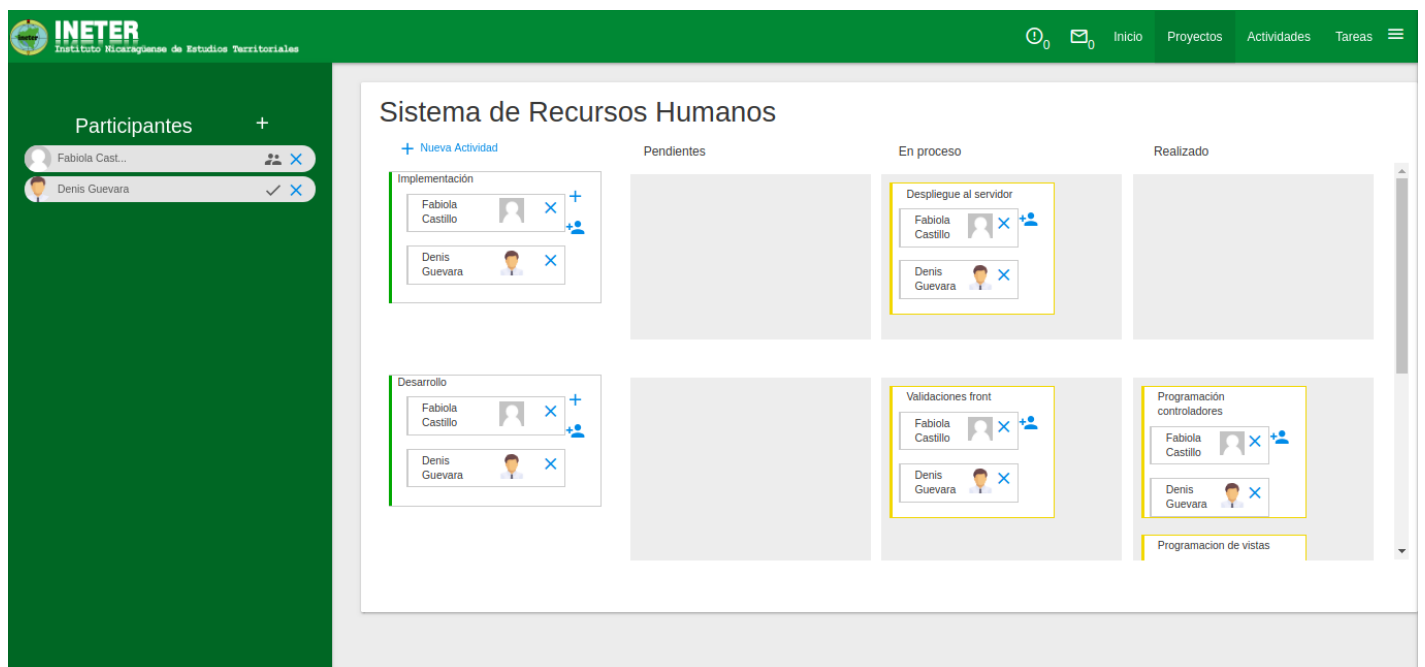


Figura 55 vista mostrar proyectos v3.3

9.7.20. Anexo 7.20: formulario editar reunión versión 3.3


The 'Editar reunión' form includes the following sections:




- Meeting Details:** A table with columns for Nombre, Fecha, Hora Inicio, Hora Fin, and Prioridad. The data row shows: Reunión Sistema en Línea Artefactos de, 17-05-2017, 13:30 PM, 16:00 PM, and Baja.
- Observación:** A text area containing a description of the meeting and a list of items to be discussed.
- Opcionales:** A section for optional items to be discussed.
- Usuarios Seleccionados:** A section for selecting users, showing a list of selected users (Dustin, Binia P..., Iván Ur..., Fabiola..., Gerald..., Francis..., Francis...) and a search bar.



Buttons for 'ATRÁS' and 'GUARDAR' are located at the bottom of the form.

Figura 56 formulario editar reunion v3.3

9.7.21. Anexo 7.21: vista mostrar reunión

**INETER**
Instituto Nicaragüense de Estudios Territoriales

InicioProyectosActividadesTareas









Reunión Sistema en Linea Artefactos de ingeniería de software

Fecha	Hora Inicio	Hora Fin
17-05-2017	01:30 PM	04:00 PM
Prioridad	Local	Estado
Alta		Realizada


Descripción


Para Certificado Catastral y licencia catastral. Se mostró el avance de artefactos de ingeniería de software. 1. Descripción general, Texto Libre. 2. Diagrama de casos de uso 3. Flujo de eventos de casos de uso 4. Lógica de Negocio. 5. Diagrama de colaboración, Diagrama de comunicación. 6. Diagrama de secuencia Faltan 7. Diagrama de actividad de la responsabilidad principal de la clase 8. Diagrama de Entidades. Persistencia de la Lógica 9. Diagrama MOR- E/R Opcionales; Diagrama de despliegue Diagrama de objetos Diagrama de componentes.

Participantes

Francisco MoraDustinFabiola CastilloGerald GuevaraBinia PadillaIván Urbina

2 Comentarios

**Fabiola Castillo**
Mandar la información
3 minutos | [Responder](#) | [Edit](#) | [2 Favoritos](#)

**Denis Guevara**
Gracias por participar de la reunión chicos
2 minutos | [Responder](#) | [Edit](#) | [1 Favorito](#)

Deja tu comentario

ENVIAR

Figura 57 vista mostrar reunion v3.3

9.8. Anexo 8: Plan informe de trabajo DSGI


Avances de Trabajo Enero - Mayo 2016																		
Dirección de Desarrollo de Software Geo-Informático (en funciones)																		
Dirección General de Sistemas Básicos (DGSE)																		
INETER																		
																		
No.	Actividades	Recursos Humanos	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Prioridad	Estado	% Porcentaje	Observación
1	Implementación IDE-Sensores	(4) MY + GG + CM + GL																
	Diseño e implementación de Plataforma Interna	MY + GG / CM																
	1.1.- Análisis y diseño de la base de datos														ALTA	Finalizado	100%	
	1.1.1.- Elaboración del diagrama de clase														ALTA	Finalizado	100%	
	1.1.2.- Elaboración del diagrama E-R														ALTA	Finalizado	100%	
	1.1.3.- Elaboración de requerimientos técnicos														ALTA	Finalizado	100%	
	1.1.4.- Diseño conceptual y lógico de la base de datos.														ALTA	Finalizado	100%	
	1.1.5.-Elaboración de la documentación del proceso														ALTA	Finalizado	100%	
	Análisis de modelado	MY + GG / CM																
	1.2.1.- Diccionario de datos														ALTA	Finalizado	100%	
	1.2.2.- Medios de almacenamiento existentes														ALTA	Finalizado	100%	
	1.2.3.- Elaboración del documento														ALTA	Finalizado	100%	
	Diseño de arquitectura de la base de datos	MY + GG / CM																
	1.3.1.- Base de datos, usuario, roles, esquemas														ALTA	Finalizado	100%	
	1.3.2.-Seguridad y redes														ALTA	Finalizado	100%	
	1.3.3.-Carga de trabajo en los servidores														ALTA	Finalizado	100%	
	1.3.4.-Elaboración del documento														ALTA	Finalizado	100%	
	Diseño relacional de la base de datos	MY + GG / CM																
	1.4.1.- Normalización de tablas														ALTA	Finalizado	100%	
	1.4.2.- Implementación de la norma UNE 500540														ALTA	Finalizado	100%	
	1.4.3.-Diccionario de datos														ALTA	Finalizado	100%	
	1.4.4.- Elaboración del documento														ALTA	Finalizado	100%	
	Mapeo conceptual y lógico de la base de datos	MY + GG / CM																
	1.5.1.- Importación e integración de los datos de distintas fuentes														ALTA	Finalizado	100%	
	1.5.2.- Aplicación de los estándares OGC e ISO 19110														ALTA	Finalizado	100%	
	1.5.3.- Elaboración de la documentación														ALTA	Finalizado	100%	
	Análisis preliminar de los requerimientos funcionales	MY + GG / CM																
	1.6.1 - Requerimientos funcionales														ALTA	Finalizado	100%	
	1.6.2 - Requerimientos No funcionales														ALTA	Finalizado	100%	
	1.6.3.1.- Caso de Uso de Lógica del Negocio	MY + GG / CM													ALTA	Finalizado	100%	
	1.6.3.2.- Caso de Uso de la Plataforma de Interoperabilidad Agroclimática.														ALTA	Finalizado	100%	
	1.6.3.3.- Caso de Uso. Gestionar Valor de la Variable														ALTA	Finalizado	100%	
	1.6.3.4.- Caso de Uso. Gestionar Estación														ALTA	Finalizado	100%	
	1.6.3.5.- Caso de Uso Gestionar Usuario														ALTA	Finalizado	100%	
	1.6.3.6.- Caso de Uso Gestionar Reporte														ALTA	Finalizado	100%	
	1.6.3.7.- Formatos de Origen de Datos														ALTA	Finalizado	100%	
	Diseño de procesos de migración de Bases de Datos	MY + GG / CM																
	1.7.1.- Base de Datos XConnect a CAELUS														ALTA	Finalizado	100%	
	1.7.2.- Diagrama de Secuencia Migración XConnect - CAELUS														ALTA	Finalizado	100%	
	1.7.3.- Diagrama de Actividades Migración XConnect - CAELUS														ALTA	Finalizado	100%	
	1.7.4.- Diseño del proceso de migración de CAELUS a 52North														ALTA	Finalizado	100%	
	1.7.5.- Diagrama de Secuencia. Migración BD CAELUS - SOS														ALTA	Finalizado	100%	
	1.7.6.- Diagrama de Actividades. Migración BD CAELUS - SOS														ALTA	Finalizado	100%	
	1.7.7.- Integración de los Datos a la Base de Datos														ALTA	Finalizado	100%	
	1.7.1.- Base de Datos XConnect a CAELUS														ALTA	Finalizado	100%	
	Diseño de los Casos de Uso	MY + GG / CM													ALTA	Finalizado	100%	
	Diseño de una arquitectura de Sistemas	MY + GG / CM																
	1.8.1.- Front-end														ALTA	Finalizado	100%	
	1.8.2.- Back-End														ALTA	Finalizado	100%	
	1.8.3.- Middleware														ALTA	Finalizado	100%	
	Diseño del esquema para la generación automática de versiones rasterizadas	MY + GG / CM													ALTA	Finalizado	100%	
	Implementación física en PostgreSQL del esquema de BD estándar Sensor Web Enablement (SWE)	MY + GG / CM													ALTA	Finalizado	100%	
	Desarrollo (Programación) de la importación e integración de los datos provenientes desde las estaciones meteorológicas automáticas hacia la BD estándar implementada.	MY + GG / CM													ALTA	Finalizado	100%	
	1.9.- Implementación de plataforma externa	C1 + C2 / K + CM													ALTA	No iniciado	0%	
	1.5.- Clientes Web	C1 + C2 / K + CM													ALTA	No iniciado	0%	
	1.9.- Documentación	C1 + C2 / GE													ALTA	Finalizado	100%	
2	Implementación IDE- Índices / Indicadores (MARENA)	(6) K + CM + GL + GE + BI + IV																
	2.1.- Revisión del estado del arte	GE / GL + IV + BI																
	2.1.1.- Auto documentación de cada uno de los indicadores a capturar en el sistema														ALTA	Finalizado	100%	
	2.1.2.- Revisión de bibliografías orientadas a indicadores														ALTA	Finalizado	100%	
	2.1.3.- Búsqueda online de sistemas y ejemplos de sistemas de monitoreo de cambio climático														ALTA	Finalizado	100%	
	2.2.- Análisis del sistema de monitoreo cambio de climático	GE / GL + IV + BI																
	2.2.1.- Casos de uso del sistema														ALTA	Finalizado	100%	
	2.2.1.- Diagrama de actividades														ALTA	Finalizado	100%	
	2.2.2.- Diagrama de despliegue														ALTA	Finalizado	100%	
	2.2.3.- Diagrama de interfaz														ALTA	Finalizado	100%	
	2.3.- Modelado del giro del negocio	GE / GL + IV + BI																
	2.3.1.- Diagrama de clases														ALTA	Finalizado	100%	
	2.3.2.- Diagrama de objetos														ALTA	Finalizado	100%	
	2.4.- Modelado de las entidades de dominio	GE / GL + IV + BI													ALTA	Finalizado	100%	
	2.4.1.- Diagrama entidad-relación														ALTA	Finalizado	100%	
	2.4.2.- Mapeo relacional para la base de datos														ALTA	Finalizado	100%	
	2.5.- Implementación de la base de datos geoespacial[relación actividad 4]	K + CM + GL + GE + BI + IV																
	2.5.1.- Implementación de base de datos estandarizada bajo el estándar O&M														ALTA	No iniciado	0%	Aplazado hasta nuevo aviso. 2017
	2.6.- Implementación de captura de datos, reportes tabulares y gráficos	K + CM + GL + GE + BI + IV																
	2.6.1.- Programación del sistema de monitoreo de cambio climático para digitación de datos y reportes														ALTA	Finalizado	100%	
	2.7.- Implementación de robots web para captura de datos de Org.	K + CM + GL + GE + BI + IV																

Figura 58 Documento actual de planificación

9.9. Anexo 9: Diagrama físico de red

9.10. Anexo 10: Carta de aceptación

10. Bibliografía

- Babich, P. (1998). *Hoshin Handbook*. California: Total Quality Engineering Inc.
- Batista, D. G. (2010). *METODOLOGÍA PARA LA EVALUACIÓN DEL SISTEMA DE CONTROL INTERNO*.
- Carreto, J. (2008, Julio 1). *Planeación Estratégica*. Retrieved from <http://planeacion-estrategica.blogspot.com/>
- EcuRed. (n.d.). *Aplicacion web*. Retrieved from Ecu Red: https://www.ecured.cu/Aplicaci%C3%B3n_web
- EcuRed. (n.d.). *Arquitectura Cliente Servidor*. Retrieved from Ecu Red: https://www.ecured.cu/Arquitectura_Cliente_Servidor
- EcuRed. (n.d.). *Bases de datos*. Retrieved from EcuRed: https://www.ecured.cu/Base_de_Datos
- EcuRed. (n.d.). *DNS*. Retrieved from EcuRed: <https://www.ecured.cu/DNS>
- EcuRed. (n.d.). *Servidor Web*. Retrieved from Ecu Red: https://www.ecured.cu/Servidor_Web
- EcuRed. (n.d.). *Sistema Gestor de Base de Datos* . Retrieved from EcuRed: https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos
- EcuRed. (n.d.). *Virtualización* . Retrieved from Ecu Red: <https://www.ecured.cu/Virtualizaci%C3%B3n>
- Eftimov, I. (2016, 04 28). *Semaphore*. Retrieved from How to Test Rails Models with RSpec: <https://semaphoreci.com/community/tutorials/how-to-test-rails-models-with-rspec>
- Enterprise Architect. (n.d.). *Manual de usuario* . Retrieved from Docplayer : <http://docplayer.es/1170834-Enterprise-architect.html>
- Grijalva, N. (2012, 10 15). *Ingeniería de software 1*. Retrieved from <http://software1nathalygrijalva.blogspot.com/2012/10/modelo-espinal.html>
- James Rumbaugh, G. B. (2007). *El Lenguaje Unificado de Modelado Manual de referencia*. Madrid: Pearson.
- Lewis. (1994). *What is Software Engineering?* Retrieved from Angelfire.
- Martínez, S. (n.d.). *Mundo Erp*. Retrieved from Ventajas e inconvenientes del software libre y Open Source (OSS) en la gestión empresarial:

- <http://mundoerp.com/blog/ventajas-e-inconvenientes-del-software-libre-open-source-oss/>
- Mestras, J. P. (2008, 09). Estructura de las Aplicaciones Orientadas a Objetos (MVC). Universidad Complutense Madrid, Dep. Ingeniería del Software e Inteligencia Artificial, España, Madrid.
- Passenger. (n.d.). *Installing Passenger + Nginx*. (on Debian 8 (with APT)) Retrieved from <https://www.phusionpassenger.com/library/install/nginx/install/oss/jessie/>
- Pavón, J. (2008). *Facultad de informática*. Retrieved from Universidad Complutense Madrid: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>
- Phusion Passenger. (n.d.). *Phusion Passenger*. Retrieved from github: <https://github.com/phusion/passenger>
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. McGraw Hill.
- Pusher. (n.d.). *What are webSockets?* Retrieved from <https://pusher.com/websockets>
- RailsGuides. (n.d.). *Fundamentos de active record*. Retrieved from GuíasRails: http://www.guiasrails.es/active_record_basics.html
- Relish. (n.d.). Retrieved from Controller specs: <https://relishapp.com/rspec/rspec-rails/docs/controller-specs>
- Ruby Org. (2017, 09). *A programmer's best friend*. (nagachica) Retrieved from <https://www.ruby-lang.org/en/news/2017/09/14/ruby-2-4-2-released/>
- Ruby Org. (n.d.). *Ruby*. Retrieved from el mejor amigo de un desarrollador: <https://www.ruby-lang.org/es/>
- Sinnexus. (2010, Agosto 5). Retrieved from http://www.sinnexus.com/business_intelligence/plan_operativo_anual.aspx
- UCINF. (2000, 03 23). *Sistemas de información*. Retrieved from <http://jms.caos.cl/si/index.html>
- Vladimir Gutierrez, J. U. (2007). Trabajo monográfico. *Universidad Nacional de Ingeniería*.

11. Índice de ilustraciones

Figura 1 Organigrama de INETER	5
Figura 2 Modelo en espiral	20
Figura 3 Espiral sistema hoja de tiempo	22
Figura 4 Diagrama de inicio de sesión	33
Figura 5 Caso de uso recuperación de contraseña	34
Figura 6 Caso de uso registrar empleado	36
Figura 7 Caso de uso general	37
Figura 8 Caso de uso proyectos.....	40
Figura 9 Caso de uso actividades	42
Figura 10 Caso de uso tareas	44
Figura 11 Caso de uso historial de tareas.....	46
Figura 12 Caso de uso de informes	48
Figura 13 Diagrama de clases.....	50
Figura 14 Diagrama de secuencia inicio de sesión	52
Figura 15 Diagrama de secuencia recuperar contraseña.....	53
Figura 16 Diagrama de secuencia registrar empleado.....	54
Figura 17 Diagrama de secuencia general.....	55
Figura 18 Diagrama de secuencia historial de tarea	56
Figura 19 Diagrama de secuencia gestionar reuniones	57
Figura 20 Diagrama de secuencia ver informes	58
Figura 21 Diagrama de despliegue	59
Figura 22 Estructura de código de la aplicación.....	62
Figura 23 factory para login usuario	66
Figura 24 pruebas para modelo proyecto.....	66
Figura 25 prueba para crear proyecto	67
Figura 26 salida de pruebas	68
Figura 27 configuración de prueba de estrés	79
Figura 28 url configuración	79
Figura 29 Ejecución de prueba de estrés	80
Figura 31 antes de ejecutar la prueba	80

Figura 31 Durante la prueba.....	80
Figura 32 Espectro de tiempo de click	81
Figura 33 Tiempo de protocolo.....	82
Figura 34 Ancho de banda de servidor y usuario	82
Figura 35 Recursos del servidor.....	83
Figura 36 Diseño de la base de datos.....	89
Figura 37 vista tareas v1.1	117
Figura 38 vista listar proyectos v2.1	118
Figura 39 vista calendario v2.1.....	119
Figura 40 formulario editar proyecto v2.1	120
Figura 41 vista mostrar proyecto v2.1	120
Figura 42 vista lista de proyectos v2.2	121
Figura 43 vista de mostrar proyecto v2.2	121
Figura 44 vista de reuniones v2.2	122
Figura 45 vista calendario v3.1.....	123
Figura 46 vista de organigrama v3.1	123
Figura 47 vista mostrar proyecto v3.1	124
Figura 48 últimos avances registrados v3.2	124
Figura 49 vista de informes v3.2	126
Figura 50 vista filtrar avance de tarea v3.2.....	127
Figura 51 vista invalidar tarea v3.2.....	127
Figura 52 vista validar tarea v3.2	128
Figura 53 vista de mensajes v3.3.....	128
Figura 54 vista calendario v3.3.....	129
Figura 55 vista mostrar proyectos v3.3	130
Figura 56 formulario editar reunion v3.3.....	130
Figura 57 vista mostrar reunion v3.3	131
Figura 58 Documento actual de planificación.....	132

12. Índices de tablas

Tabla 1 Costos de desarrollo.....	31
Tabla 2 riesgos y contingencias	64
Tabla 3 Razón de desempeño	76
Tabla 4 Facilidad de uso	77
Tabla 5 Pasos de procedimiento	77
Tabla 6 Tiempos para reporte anual	78
Tabla 7 Entrevista Dir. Desarrollo.....	86
Tabla 8 Entrevista Dir. Geo repositorio	87
Tabla 9 Tabla actividades.....	90
Tabla 10 Tabla ar_internal_metadata.....	91
Tabla 11 Tabla comment_reunions	91
Tabla 12 Tabla comments	92
Tabla 13 Tabla likes	93
Tabla 14 Tabla local_managements	94
Tabla 15 Tabla locals	95
Tabla 16 Tabla logged_actions	95
Tabla 17 Tabla messages	97
Tabla 18 Tabla notifications.....	98
Tabla 19 Tabla proyectos	98
Tabla 20 Tabla projects_directions.....	100
Tabla 21 Tabla registration_tokens	101
Tabla 22 Tabla replies.....	102
Tabla 23 Tabla reunions.....	102
Tabla 24 Tabla reunions_users	104
Tabla 25 Tabla roles.....	104
Tabla 26 Tabla schema_migrations	106
Tabla 27 Tabla technologies	106
Tabla 28 Tabla users.....	107
Tabla 29 Tabla work_histories.....	109
Tabla 30 Tabla works	110
	139

